

**MODIFICATION AND DESIGN OF
NON-LINEAR SYSTEM
BEHAVIOUR USING FUZZY LOGIC
AND DESCRIBING FUNCTION
METHODS**

by

George Page

A thesis submitted in partial fulfilment of
the requirements of Liverpool John Moores
University for the degree of

Doctor of Philosophy

2011

ACKNOWLEDGEMENTS

The author wishes to thank his supervisors, Dr Barry Gomm and Dr Steve Douglas for their support and guidance throughout the course of this study. They have been pillars of strength and sources of tremendous insight and I have been fortunate to have had such superb supervision. I feel deeply indebted to them and they have my sincere thanks.

Also, I wish to thank Professor Ian Jenkinson for allowing me to undertake this doctorate and for providing the facilities and for his continued support.

My sincere thanks must also go to my wife, Mavis, who has supported me so magnificently and put up with my foibles so uncomplainingly during the course of this work.

ABSTRACT

Whilst it is a simple matter to outline the salient features of any non-linearity, predicting the way in which it will affect the response of a system to incoming signals is not so straightforward. Furthermore, removing any undesirable effects, such as limit-cycling, is often so difficult that systems which exhibit these effects are run well below the desired operating conditions simply to avoid them coming into play. To completely remove the effects of a non-linearity was not considered to be possible. This thesis shows that it is indeed possible to completely remove non-linear effects and, along the way, presents new algorithms which streamline the whole process of non-linear analysis and design.

The thesis begins by outlining the basic characteristics of non-linear systems and the ways in which they can be analysed. The methods for deriving describing functions have been investigated and a new algorithm formulated which enables the rapid delineation of the entire family of real describing functions. The use of this algorithm enables a comprehensive range of non-linearities to be designed and simulated and their behaviours analysed.

However, the algorithm also enables the describing functions of more complicated non-linear systems to be calculated for which models are not readily obtainable. Fuzzy logic techniques are developed which can mimic these more complicated systems. These more complicated fuzzy non-linearities are simulated and analysed in detail and a variety of behaviour-patterns seen to emerge which are not so obvious when only the simpler systems are investigated.

The fuzzy logic techniques have proved to be sufficiently flexible to enable non-linear systems to be developed which are able to cancel out the effects of other non-linearities. The behaviour of these new functions has been investigated and an algorithm developed which enables 'inverses' of all of the real non-linearities to be created, an inverse non-linear function being described as one which completely nullifies the effect of another nonlinearity. Further investigation of these completely new functions shows that the effects of non-linearities and their inverses are often mutual.

TABLE OF CONTENTS

1.	Introduction	1
1.1	Background	1
1.1.1	Modelling real systems	1
1.1.2	Non-linear systems	2
1.1.3	Fuzzy-logic control systems	4
1.2	Research objectives	5
1.3	Structure of the thesis	7
2.	Characteristics of Non-linear Systems	10
2.1	Introduction	10
2.2	Basic Characteristics	10
2.3	Common types of non-linearity	13
2.3.1	Continuous nonlinearities.....	13
2.3.2	Discontinuous non-linearities.....	14
2.3.2.1	Saturation.....	14
2.3.2.2	Dead zone (or dead-space).....	15
2.3.2.3	The ideal relay.....	16
2.3.2.4	An ideal relay with dead zone.....	17
2.3.2.5	Friction.....	17
2.4	Conclusion.....	19
3.	Methods of Analysing Non-linear Systems	20
3.1	Introduction	20
3.2	The Phase-Plane Approach.....	21
3.3	The Describing Function Approach.....	22
3.4	The Describing Function Method in Detail	23
3.5	Stability Analysis Using Describing Functions.....	25
3.6	Conclusion.....	31
4.	A General Solution for a Family of Real Describing Functions.....	33
4.1	Introduction.....	33
4.2	The Graphical Method	34
4.3	A General Solution.....	36
4.4	An Algorithm for Generating Real Describing Functions	40
4.5	Some Typical Examples	41
4.5.1	The trivial case: the straight line	42
4.5.2	Hard saturation.....	42
4.5.3	Soft saturation.....	43
4.5.4	Dead-zone	44
4.5.5	Dead-zone plus hard saturation	45
4.5.6	Dead-zone plus soft saturation.....	46
4.5.7	The ideal relay (or pure Coulomb friction).....	47
4.5.8	Coulomb friction plus viscous drag.....	48
4.5.9	Coulomb friction plus viscous drag plus saturation	48
4.5.10	Triple slope non-linear characteristics.....	49
4.5.11	A non-linearity with three break-points (four slopes)	51
4.5.12	The four-breakpoint case.....	52

4.6	Continuous Real Non-linear Systems	54
4.7	Conclusion	55
5.	Simulation and Analysis of Non-linear Systems	57
5.1	Introduction	57
5.2	Testing the system with the Standard Non-linearities.....	60
5.2.1	Hard saturation	60
5.2.2	Soft saturation	63
5.2.3	Dead-zone	64
5.2.4	Dead-zone plus hard saturation.....	65
5.2.5	Dead-zone plus soft saturation.....	67
5.2.6	The ideal relay (or pure Coulomb friction)	68
5.2.7	Coulomb friction plus viscous drag.....	70
5.2.8	Coulomb friction plus viscous drag plus saturation	73
5.2.9	Triple-slope non-linear characteristics.....	75
5.3	Linearities which produce more than one limit-cycle	77
5.3.1	A non-linearity with three break points (four slopes)	78
5.3.2	The three four-breakpoint case	79
5.4	The significance of critical points.....	80
5.5	Conclusions	80
6.	Fuzzy Systems	82
6.1	Introduction.....	82
6.2	Fuzzy control systems.....	83
6.3	Methods of tuning fuzzy control systems.....	85
6.4	The tuning approach chosen	88
6.4.1	The fuzzy input sets	89
6.4.2	The choice of inference method	89
6.4.3	Possible use of an inference filter.....	90
6.5	The fuzzy approach used in this research	91
6.6	A Template	92
6.7	Conclusion	95
7.	Simulation and Analysis of Fuzzy Non-linear Systems	96
7.1	Introduction.....	96
7.2	Fuzzy Hard Saturation	98
7.3	The Dead-zone Non-linearity.....	100
7.4	The relay (or Coulomb friction).....	102
7.4.1	Coulomb friction plus viscous drag.....	102
7.4.2	Coulomb friction plus viscous drag plus hard saturation.....	104
7.5	The triple slope non-linear characteristic.....	105
7.5.1	Case (i): $K_0 > K_1 > K_2$	106
7.5.2	Case (ii): $K_0 > K_1$ & $K_2 > K_1$	107
7.5.3	Case (iii) - $K_0 < K_1$ & $K_2 < K_1$	110
7.6	Linearities which can produce more than one limit-cycle.....	112
7.6.1	The three break-point case	113
7.6.2	The four break-point case.....	115
7.7	Discussion of Results	118
7.8	Conclusion.....	122

8.	Using Fuzzy Methods to Modify and Control Non-linear Behaviour	124
8.1	Introduction.....	124
8.2	The effect of an inverse function.....	125
8.3	The design of a fuzzy non-linearity and its inverse.....	128
8.4	Proving the method.....	130
8.4.1	A block-diagram explanation of how the design method was tested	130
8.4.2	The range of design features on which the method was tested	133
8.5	Some Standard Results.....	133
8.5.1	Dead-zone plus soft saturation.....	134
8.5.2	Coulomb friction plus viscous drag.....	135
8.5.3	The triple-slope non-linearity	141
8.6	The double limit-cycle examples	145
8.6.1	The three break-point (four-slope) non-linearity	145
8.6.1.1	The complete inverse function.....	146
8.6.1.2	Partial inverse functions.....	148
	(i) Removal of the higher limit-cycle	148
	(ii) Removal of the lower limit-cycle.....	151
8.6.2	The four breakpoint case	153
8.6.3	Continuous real non-linear systems.....	158
8.6.3.1	A continuous polynomial which exhibits saturation	158
8.6.3.2	A continuous polynomial which does not exhibit saturation	162
8.7	Some general comments.....	168
8.8	Conclusions	170
9.	Conclusions and suggested further work	171
9.1	Conclusions.....	171
9.2	Suggested further work.....	174
	References	178
	Appendices	
A1	Describing functions of real, not complex, systems.....	185
A1.1	A general solution for the family of real describing functions	186
A1.2	Matlab code for the general solution with four breakpoints	191
A1.3	Describing function for Coulomb friction plus viscous drag plus saturation	194
A2	Inverse Nyquist Calculations	196
A2.1	Inverse Nyquist calculations	197
A2.2	Coding for selected describing functions	200
A3	Inverse Functions and sample SIMULINK models.....	212
A3.1	Relation between the slope of a function and its mirror image	213
A3.2	Algebraic relation between dead-zone and Coulomb friction.....	214
A3.3	The design of a fuzzy non-linearity and its inverse	216
A3.4	Sample SIMULINK models.....	218
A3.5	A Template for the Design of the Fuzzy Non-linear Functions ...	221

LIST OF FIGURES

Figure No.	Title	Page
2.1(a)	The hard-spring case	13
2.1(b)	The soft-spring case	13
2.2(a)	The hard saturation case	15
2.2(b)	The soft saturation case	15
2.3	Dead-zone	16
2.4	The ideal relay	16
2.5	Ideal relay plus dead-zone	17
2.6(a)	Theoretical frictional effects	18
2.6(b)	Combined frictional effect	18
3.1	Basic Linear system with unit feedback	25
3.2	System with nonlinearity shown separately	26
3.3	Inverse Nyquist/describing function graphs	28
3.4	More inverse Nyquist/describing function graphs	30
3.5	A real describing function locus with an inverse Nyquist magnitude superimposed on it	31 and 58
4.1	The basic describing function graphical approach	36
4.2	Non-linear function with n breakpoints	38
4.3	Cut-off points on the sine curve	38
4.4	Relationship between P and Q	40
4.5	The straight line and its describing function	45
4.6(a)	Hard saturation non-linearity	43
4.6(b)	Hard saturation describing function	43
4.7(a)	Soft saturation non-linearity	44
4.7(b)	Soft saturation describing function	44
4.8	Dead-zone non-linear characteristic and its describing function	45
4.9	Dead-zone plus hard saturation	46
4.10	Dead-zone plus soft saturation	46

4.11	Relay (Coulomb friction)	47
4.12	Coulomb friction plus viscous drag	48
4.13	Coulomb friction plus viscous drag plus saturation	49
4.14(a)	The triple-slope nonlinearity	50
4.14(b)	Describing functions for various combinations of triple slopes	50
4.15(a)	Non-linearity with three breakpoints	52
4.15(b)	Describing function for the three breakpoint case	52
4.16(a)	Non-linearity with four breakpoints	53
4.16(b)	The describing function for the four-breakpoint case	54
5.1	Impulse response of standard transfer function with unit feedback	58
5.2	Basic testing arrangement	59
5.3	Limit-cycle oscillation with hard saturation	61
5.4(a)	Circuit to display the non-linear characteristic	61
5.4(b)	Hard Saturation characteristic	62
5.5	Plot of describing function $N(x)$ against input x – hard saturation	62
5.6	Plot of describing function $N(x)$ against input x – soft saturation	64
5.7	Simulation arrangement for dead-zone plus hard saturation	65
5.8	Limit-cycle oscillation with hard saturation and dead-zone	66
5.9	Dead-zone plus saturation characteristic	66
5.10	Plot of describing function of dead-zone plus hard saturation	67
5.11	Describing function $N(x)$ against input x – dead-zone plus soft saturation	68
5.12	Describing function for pure Coulomb friction	69
5.13	Coulomb characteristic	69
5.14	Plot of describing function for Coulomb friction (ideal relay)	70
5.15(a)	Coulomb friction plus viscous drag > inverse Nyquist cross-over value	71
5.15(b)	Coulomb friction plus viscous drag < inverse Nyquist cross-over value	71

5.16	Coulomb friction plus viscous drag characteristic	72
5.17	Plot of describing function for Coulomb friction plus viscous drag	72
5.18	Coulomb friction plus viscous drag plus hard saturation	73
5.19	Coulomb friction + viscous drag + saturation characteristic	74
5.20	Plot of describing function for Coulomb friction + viscous drag + saturation	74
5.21	Describing function for the triple-slope characteristic – case (i)	75
5.22	Describing function for the triple-slope characteristic – case (ii)	76
5.23	Describing function for the triple-slope characteristic – case (iii)	76
5.24	Describing function for the three break-point characteristic	78
5.25	Describing function for the four break-point characteristic	79
6.1	Fuzzy input sets representing temperature range	84
6.2	A typical fuzzy control arrangement	85
6.3	Use of an inference filter	87
6.4	A Template for the design of the fuzzy non-linear functions	93
7.1	Testing arrangements to ensure the non-linearities have the correct parameters	97
7.2	Settings for the fuzzy hard saturation module	99
7.3	General SIMULINK model of the fuzzy non-linear control circuits	99
7.4	Fuzzy hard saturation impulse responses	100
7.5	Settings for the fuzzy dead-zone + saturation model	101
7.6	Fuzzy dead-zone + saturation response	102
7.7	Settings for Coulomb friction plus viscous drag model	103
7.8	Fuzzy Coulomb friction plus viscous drag response	103
7.9	Settings for Coulomb friction plus viscous drag plus saturation	104
7.10	Fuzzy Coulomb friction plus viscous drag plus	

	saturation response	105
7.11	Settings for the fuzzy triple-slope (Case (i))	106
7.12	Fuzzy triple-slope response	107
7.13	Triple-slope describing function showing cross-over values	108
7.14	Settings for the fuzzy triple-slope model – case (ii)	109
7.15	Fuzzy triple-slope response case (ii)- gain of 50	109
7.16	Triple-slope response case (iii) with step input less than the critical value	110
7.17	Triple-slope response case (iii) with step input greater than the critical value	111
7.18	Fuzzy triple-slope response case (iii) with step input greater than the limit-cycle value	111
7.19	Settings for the fuzzy triple-slope model – case (iii)	112
7.20	Settings for the fuzzy three-breakpoint model	114
7.21	Fuzzy three-breakpoint response showing two limit-cycles	114
7.22	Settings for the fuzzy four-breakpoint model	115
7.23	Fuzzy four-breakpoint response showing two limit-cycles	116
7.24	Fuzzy four-breakpoint response showing two limit-cycles and runaway oscillation	117
7.25	Limit-cycle spontaneously flipping from the higher value to the lower	117
7.26(a)	Effects of relative positions of limit-cycles and critical points – stable system	120
7.26(b)	Effects of relative positions of limit-cycles and critical points – runaway oscillation if input signal is too high	121
7.26(c)	Effects of relative positions of limit-cycles and critical points – runaway oscillation if the higher limit-cycle is initiated	121
8.1	The effect of a perfect inverse function	125
8.2	Dead-zone plus linearity with its mirror image in the unit ramp	126
8.4	General geometric design of a fuzzy non-linearity	

	and its inverse	130
8.6	Open-loop arrangement of non-linearity and its inverse	131
8.7	Feedback arrangement for the transfer function alone	131
8.8	Feedback arrangement with the non-linearity present	132
8.9	Inverse cancelling-out the effect of the original non-linearity	132
8.10	Describing function characteristic dead-zone+soft saturation	136
8.11	Fuzzy dead-zone plus soft saturation model and its inverse	137
8.12	Ramp input to the inverse and then dead-zone plus soft saturation in that order (open loop)	138
8.13	Output with dead-zone + soft saturation + standard transfer function	138
8.14	Step response to inverse + dead-zone in series with standard transfer function	139
8.15	Ramp input to Coulomb friction and its inverse, (open-loop). Inverse between ramp input and the Coulomb friction	139
8.16	Closed-loop step-response to the inverse function + Coulomb non-linearity in series with the standard transfer function	140
8.17	Coulomb + inverse in series with standard transfer function with the inverse function placed after the transfer function	141
8.18	Results of the triple-slope non-linearity and its inverse in series with the ramp input (open loop)	142
8.19	The triple-slope model and its inverse	143
8.20	Limit-cycle output using the triple-slope fuzzy non-linearity	144
8.21	Step response to triple-slope in series with its inverse	144
8.22	The three-breakpoint model and its complete inverse	146
8.23	Fuzzy three breakpoint response showing two limit-cycles	147
8.24	Closed-loop response of the three breakpoint model and its inverse in series with the standard transfer function	147
8.25	The describing function for the three breakpoint inverse	148
8.26	The first modification to the inverse describing function which removes the higher limit-cycle	149

8.27	The four-slope (three breakpoint) response with the first modification to the inverse function which allowed run-away oscillation.	150
8.28	The second modification to the inverse which also removes the second limit-cycle but keeps the system stable	150
8.29	The three breakpoint response, upper limit-cycle removed, with the inverse keeping the system stable	151
8.30	Modified describing function to remove lower limit-cycle	152
8.31	Inverse applied to remove lower limit-cycle	152
8.32	The four break-point model and its inverse	154
8.33	The effect of the four break-point non-linearity	155
8.34	Ramp input to open-loop four break-point model and effect of its inverse	155
8.35	The effect of the four break-point (five-slope) non-linearity cancelled out by its inverse	156
8.36	Modified non-linearity, four break-point (five-slope) with final slope value set to 0.1	157
8.37	The describing function for the non-linearity which produced a reversion to the lower limit-cycle	157
8.38	The cubic polynomial with a curve similar to a saturation non-linearity	158
8.39	Polynomial describing function for four breakpoints	159
8.40	The continuous non-linearity $p = -0.01125x^3 + 0.69x$ and its inverse	160
8.41	Output using the continuous non-linearity	161
8.42	Result of inverse and original signal	162
8.43	Describing function for the ninth-order polynomial	163
8.44	The continuous non-linearity, ninth-order, non-linearity and its inverse	165
8.45	Effect on the basic transfer function of the non-linearity $p = 0.0001x^9 - 0.0032x^7 + 0.0501x^5 - 0.2726x^3 + 0.9256x$	166
8.46	Effect of inverse on the original non-linearity with a ramp input and open-loop	167
8.47	Inverse in series with the 9 th -order non-linearity in series	

	with the standard test transfer function and unit feedback compared with the performance of the test transfer function on its own.	168
A1.1.1	Relationship between P and Q	190
A1.3.1	Characteristics for Coulomb friction + viscous drag + saturation	194
A2.1.1	Root locus for open-loop transfer function $g = \frac{50}{s^3 + 5s^2 + 6s + 1}$	199
A3.1.1	Relation between a slope and its reflection in the 45° axis	213
A3.2.1	Dead –zone characteristic	214
A3.2.2	Dead-zone removed by adding Coulomb friction	215
A3.3.1	Relation between P and P*	216
A3.3.2	Relation between c, K and P	217
A3.4.1	SIMULINK arrangement to determine the open-loop responses of a non-linearity and its inverse to a ramp input	218
A3.4.2	SIMULINK arrangement to show effect of a fuzzy non-linearity on the standard transfer function	219
A3.4.3	SIMULINK arrangement to show how the fuzzy inverse cancels out the effect of the original fuzzy non-linearity	220
A3.5.1	Template for the design of the fuzzy non-linear functions	221

LIST OF TABLES

8.1	General scaling transformations to create inverses	127
8.2	Effect of non-linearity	153
8.3	Effect of inverse	153
8.4	$p = -0.01125x^3 + 0.69x$	159
8.5	Inverse of table 8.4	159
8.6	Ninth-order Polynomial Data	163
8.6	Inverse of table 8.6	163
A3.2.1	Removing dead-zone	215

Chapter One

INTRODUCTION

1.1 Background

1.1.1 Modelling real systems

When trying to understand the behaviour of a physical system, mathematical models of the component parts are usually constructed. To simplify the mathematics it is usually assumed that the system is linear. This is because many of the techniques such as Bode plots, Nyquist diagrams, Nichols charts, Laplace function models, state-space models, and many more, which have had an important role to play in systems theory and design only work for systems whose transfer functions can be represented by linear polynomials. The effectiveness of a mathematical model is judged by the closeness with which its predicted behaviour represents that of the real system. If the behaviour of the model is sufficiently close to that of the real system then a linearized version can be found, taking care about the range of operating conditions for which the linearization is assumed to be valid (Atherton, 1981; Dutton *et al.* 1997). However, all real systems are nonlinear to some extent and if their dynamic behaviour cannot be adequately represented by a linear model then some method has to be found to incorporate this into the plant model to obtain suitably meaningful results. Whereas linear systems obey the principle of superposition, non-linear systems do not; a simple doubling of the magnitude of an input signal

does not necessarily produce a doubling of the magnitude of the output signal. Furthermore, a sinusoidal input to a non-linear system can lead to the production of harmonics and sub-harmonics at the output – something which can not happen with a linear system. Consequently if two systems, each of which exhibit non-linear behaviour, are included in the same feedback loop, the resultant behaviour pattern will not be the simple sum of the two contributing parts.

1.1.2 Non-linear systems and unwanted oscillatory behaviour

Certain types of non-linear system (Khalil, 2002) can exhibit a phenomenon known as limit-cycle or spontaneous oscillation and this effect has been of particular interest as far as this research has been concerned. The effect is quite distinct from ordinary resonance and is independent of the frequency content of any input signal. It is a property of the architecture of the nonlinear system itself and occurs spontaneously when the amplitude of the input signal enters a critical range. This critical range may exist at low amplitudes but not at high, at high amplitudes but not at low; or it may even exist for one or more intermediate ranges of amplitudes. When a limit cycle occurs, the affected system goes into a stable sinusoidal, or even non-sinusoidal, oscillation with an amplitude and frequency which are characteristics of the architecture of the individual system and are in no way connected to the frequency and amplitude of any inputs.

In control systems the complete system model is often determined by modelling the individual components, in which case the linear and non-linear elements may be separately described. From this realization the describing function method

gradually evolved in the late 40s and 50s (Kochenburger, 1950) and when this is combined with classical control methods e.g. Nyquist diagrams (Kochenburger, 1950; Atherton, 1981; Nagrath *et al.*, 1980) it becomes possible to predict when limit cycles may occur and also to predict their amplitudes and frequency of oscillation. However it has not proved possible with classical control techniques to easily remove or ameliorate these unwanted effects. Once a limit cycle starts, control of the affected system is lost. Changing the amplitude of the input signal will often have no effect. Even removing all input signals will usually have no effect. Once it has started, a limit cycle is self-sustaining and usually the only way to stop it is to remove all power and shut down the system.

Up to now, the only way in which the unwanted oscillations can be avoided is to make sure that the amplitudes of input or feedback systems reside outside certain critical ranges or, more drastically, to redesign the system itself. Having to use signals within limited ranges of amplitude can seriously curtail the performance of a system but sometimes it is a price that has had to be accepted. Redesigning the very architecture of a system is not always an option.

In this thesis a new method is presented which avoids the necessity of redesigning the system itself by enabling the characteristics of the non-linearity to be adjusted to either remove the conditions in which the unwanted oscillations occur or to move them into regions of error signal amplitude in which they no longer present operational problems.

1.1.3 Fuzzy logic control systems

Fuzzy logic control was developed by Mamdani and his team in the 1970s (Mamdani, 1974; Mamdani and Assilian, 1975; Mamdani and Procyk, 1979) and was first used with systems whose transfer functions could not easily be determined. Fuzzy control systems are inherently nonlinear and can themselves exhibit the range of features of classical non-linear systems. In particular they can exhibit limit-cycle oscillation (Gordillo *et al.*, 1997), although usually by accident rather than design. They can be made to model the behaviour like adaptive vehicle suspension units in which the coefficients of restitution change depending upon the previous behaviour of the system. But, again, it must be stressed that these effects have arisen more by accident than by design. The ability to create these non-linear effects using fuzzy systems has been demonstrated (Kim, *et al.*, 2000; Cuesto *et al.*, 1999) and gain and phase margin analysis of such systems has been investigated (Perng *et al.*, 2003). It is even possible to create PD and PI Fuzzy Controllers (Page *et al.*, 1999; Kwok *et al.*, 1991) and to use describing functions to analyse them (Aracil *et al.*, 2004). However, although it is possible to use a fuzzy system to create a desired limit-cycle effect (Kim *et al.*, 2000), there have been no reports of attempts to use this possibility to investigate whether using such a system as a design tool will enable the modification of the shape of a signal to control existing non-linear effects. The whole subject area of signal modification using fuzzy techniques has hardly been touched and, in particular, the ability to mimic specific non-linear effects has had very little investigation. The idea of trying to use fuzzy systems to

create inverse non-linear effects which could cancel out specific non-linearities has had no reported investigation at all. This is the very topic which has been addressed in this research. There is a well-established and published body of work using, for example, feedback linearization and computer torque techniques but they do not offer the simplicity of fuzzy systems, nor can they cope well with large non-linearities or with systems which exhibit memory (Smith *et al.*, 2000; Lim *et al.*, 2002; Kang *et al.*, 1999; Nassirharand and Karimi, 2006). Also, feedback linearization usually requires a complicated feedback control law which can be sensitive to parameter uncertainties (Tewari, 2002; Nijmeijer & van der Schaft, 1990) .

1.2 Research objectives

Fuzzy logic techniques provide a method of modifying the actual shapes of signals which is not easily achieved by other means. Furthermore, the capacity for fine-detail custom design of non-linear effects is virtually impossible by any other method. It is the investigation of this signal-shaping capability and its use to control the behaviour of non-linear systems which is at the core of this research.

The purpose of this work was firstly to develop a signal-modification system using fuzzy logic which would enable various standard non-linearities to be mimicked. To avoid confusion, only type 1 fuzzy systems have been used in this work; since using blurred fuzzy logic, type 2 systems, (Sepulveda *et al.*, 2007) has not been shown to confer any extra benefits as far as control systems are concerned. A type 1 fuzzy system is one in which the envelopes defining the fuzzy sets consist of crisp clear straight lines; in type 2 fuzzy systems the

envelopes defining the fuzzy sets are themselves fuzzy. The strengths and weaknesses of the available fuzzy logic controllers were assessed (Takagi, 1993; Castro, 1995) and the most suitable was chosen for the particular task in hand. A method was developed of using the Sugeno approach to mimic a group of standard non-linearities which did not possess memory and so produced real, as opposed to complex, describing functions. A decision was made to confine the investigation only to this group of non-linearities and the only restriction this decision imposes is that while limit-cycles of different amplitude for a given system can be investigated, the capacity to move them to different frequency ranges is lost. This was considered to be a restriction worth accepting in this research. The ability to move to different frequency ranges as well as different amplitude ranges, which would follow with complex describing functions, could be investigated in further work.

The behaviour of these fuzzy mimics, or simulations, was checked against standard non-linearities available in the SIMULINK toolbox. These standard non-linearities were used in series with a type 0 third-order system to produce a series of limit-cycles whose frequency and amplitude could be compared with those predicted by theory. The particular third-order system was chosen because it was the system used in paper which prompted this line of research (Kim, *et al*, 2000) and the intention was to compare some preliminary results with that work. In order to accomplish this, a family of describing functions was derived and a general algorithm developed so that a real describing function of any complexity could be easily obtained. Again, to the best of the author's knowledge no such algorithm has been reported in the literature. The describing functions produced using this algorithm were then used in conjunction with the inverse Nyquist plot

of the linear transfer function mentioned above. This transfer function (see equation 5.1) is dealt with in more detail in chapter five. In this way the limit-cycles could be predicted and their characteristics obtained. The next stage is really the crux of the whole research effort, the development of the non-linearities which would eliminate the existing non-linearities and so remove the limit-cycles. Effectively, the aim was to develop inverses of the existing non-linearities and their associated describing functions. In the event, from the development of adequate inverse functions for the family of real describing functions it was possible to discern patterns in their development. For this work the author used MATLAB. This proved adequate for all of the work that was carried out although the fuzzy logic toolbox was not flexible enough to enable some more sophisticated techniques to be constructed and incorporated into SIMULINK.

1.3 Structure of the thesis

In chapter two there is an overview of non-linear systems, together with a review of the relevant literature and a discussion of the various techniques that could be used for their analysis. Chapter three presents an explanation of the techniques particularly appropriate for the type of non-linear analysis considered, i.e. phase-plane descriptions and the describing function approach, together with a discussion of stability issues. There are, of course, many techniques in existence for non-linear analysis and control system design for example: Lyapunov's method, sliding mode, feedback linearization, Popov and Zames methods but, as

explained in chapter three, only two are appropriate for the geometrical approach which has been used in this thesis.

Chapter four gives a detailed description of the describing function technique. A general approach for deriving the family of real describing functions is presented and all those functions relevant to this research programme are obtained. A useful algorithm for rapidly obtaining these functions is also presented. Because of the mathematical nature of this chapter complete derivations are shown in Appendix A1.1. (This approach has been adopted throughout the thesis to avoid cluttering the main text and also as a quick resource if any derivations need to be checked.) To complete the non-linear discussion, chapter five uses the describing function results, in conjunction with the inverse Nyquist of the third-order transfer function model of the linear part of the system, to predict the occurrence, i.e. the frequencies of oscillation and the amplitudes, of any limit cycles that are produced with the various non-linearities. This is reinforced by using SIMULINK and the standard non-linearities supplied within that package to produce limit-cycle outputs for the same third-order model and then comparing them with the predictions. Wherever possible, in all of this work, the errors in these measurements are also estimated.

Chapter six follows on with a discussion of the general characteristics of fuzzy systems and the various approaches relevant to the research investigation and a discussion of the appropriate literature. Chapter seven describes how fuzzy systems can be designed which mimic non-linear behaviour. When the fuzzy non-linearities so obtained are put into the forward path in the simulation system used for this research, it is possible to compare the characteristics of the resultant

limit cycles with theoretical results obtained earlier. This work is followed by designing more complicated non-linearities to produce multiple limit-cycles which can be selectively switched on and off by suitable input signals. Chapter eight shows how fuzzy inverse functions can be derived. The effectiveness of these inverse functions is tested by placing them in series with the functions whose behaviour they are meant to cancel out. A general method is also suggested for obtaining such inverses using fuzzy logic methods. Again, to the best of the author's knowledge, these techniques have neither been published nor reported before.

The overall approach has been to present the various types of non-linearity as part of an evolving story. So the same simple nonlinear examples appear several times but in each succeeding appearance they are presented in a more developed form and other more complicated non-linear functions start to appear to demonstrate more subtle features, such as multiple limit-cycles.

Chapter nine presents a detailed discussion of the results obtained, the limitations of the method and a pointer as to how the describing functions and the fuzzy mimics of more complex systems might be obtained. Finally, the conclusions of the investigation are presented together with a discussion of, and detailed suggestions for, further work.

Chapter Two

SOME CHARACTERISTICS OF NON-LINEAR SYSTEMS

2.1 Introduction

The purpose of this chapter is to briefly explain the basic differences between linear and non-linear systems and to outline the commonest non-linearities used in this research work. Only the physical characteristics are mentioned here and only for a selection naturally occurring real non-linearities.

2.2 Basic Characteristics

Non-linear behaviour is present in all physical systems, (Atherton *et al.*, 1980; Leigh, 1983; Slotine and Li, 1991). Even linear systems will behave in a non-linear fashion if taken outside their specified operating range. Sometimes non-linear features will be deliberately introduced to improve a system's performance, to make it safer (e.g. to keep its output within a desired range) or even to make the overall performance more economical than might be the case with linear components alone.

There are several ways in which non-linear systems differ from their linear counterparts. The principle differences are:

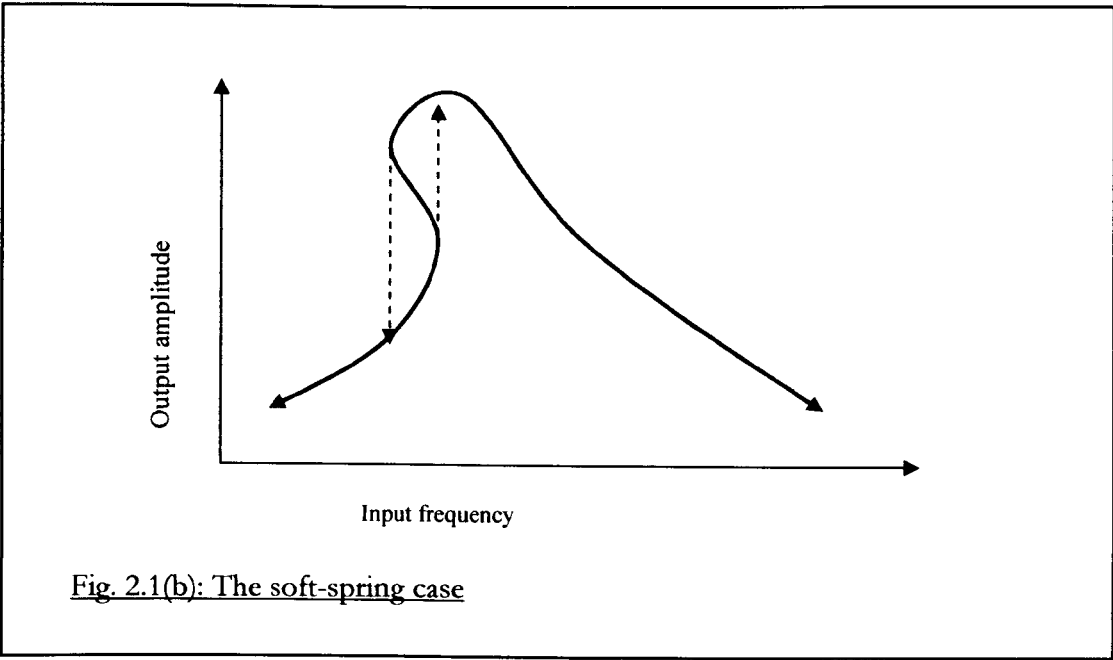
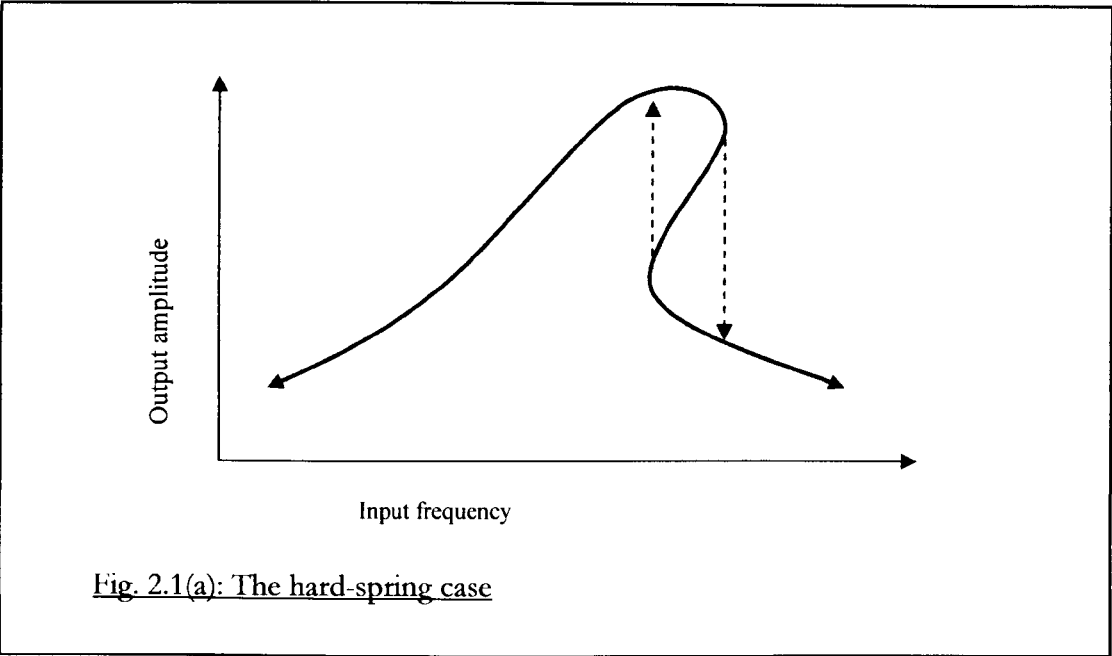
- (i) The mathematical principle of superposition no longer applies. For linear systems, altering the magnitude of the input would not change the shape of the output response, whereas for a non-linear

system there could be a considerable change both in the percentage overshoot and in the frequencies of oscillation.

- (ii) Stability behaviour of non-linear systems is different. In a linear system this is a characteristic of the system and is determined entirely by the location of the poles. It is independent of the magnitude and nature of the input. Furthermore, the application of a sinusoidal input to a stable linear system will produce a sinusoidal output of the same frequency whose only difference will be in phase and magnitude. If, on the other hand, the system was non-linear there would likely be several harmonics, or sub-harmonics present which would not have appeared in the linear case. Furthermore, the range of stable behaviour of non-linear systems is governed by effects which are completely unknown in linear systems as explained in sections (iii) and (iv) below.
- (iii) Limit-cycle oscillations cannot exist in linear systems but with non-linear systems they can spontaneously occur when the magnitude and frequency of the input signal enter certain ranges which are properties of the architecture of the system itself, in which the non-linear components are embedded. However, once they start, the magnitude and frequency of the limit-cycles will also be solely determined by this system architecture and will be independent of the input signal, or of any other initial conditions. Furthermore, the shape of the spontaneous limit cycle may not be sinusoidal. Once they have started, control of the system is lost, and the limit-cycle oscillations will continue regardless of any

input signal. As limit –cycle behaviour is a core feature of non-linear responses it was the main feature investigated with only non-linearities with real describing functions being considered. The reason why the addition of a modifying non-linearity into the forward path can control the apparently uncontrollable is that it effectively alters the architecture of the system and acts as an adaptive controller.

- (iv) Jump-resonance in non-linear systems is a phenomenon in which there are discontinuous jumps in magnitude and phase of the output as the initial input frequency is changed. The effect can take two forms: (a) the hard-spring case as shown in Figure 2.1a and (b) the soft-spring case as shown in Figure 2.1b. In both cases the input is gradually increased in frequency from zero whilst keeping its amplitude fixed; the output amplitude then varies as shown. It will be noticed that for a range of frequencies two different output amplitudes are possible, depending on whether the input frequency is rising or falling.



2.3 Common types of non-linearity

2.3.1 Continuous nonlinearities

These are elements whose input-output characteristics can be defined, (Atherton *et al.*, 1980; Dutton *et al.*, 1997), by analytical functions and whose

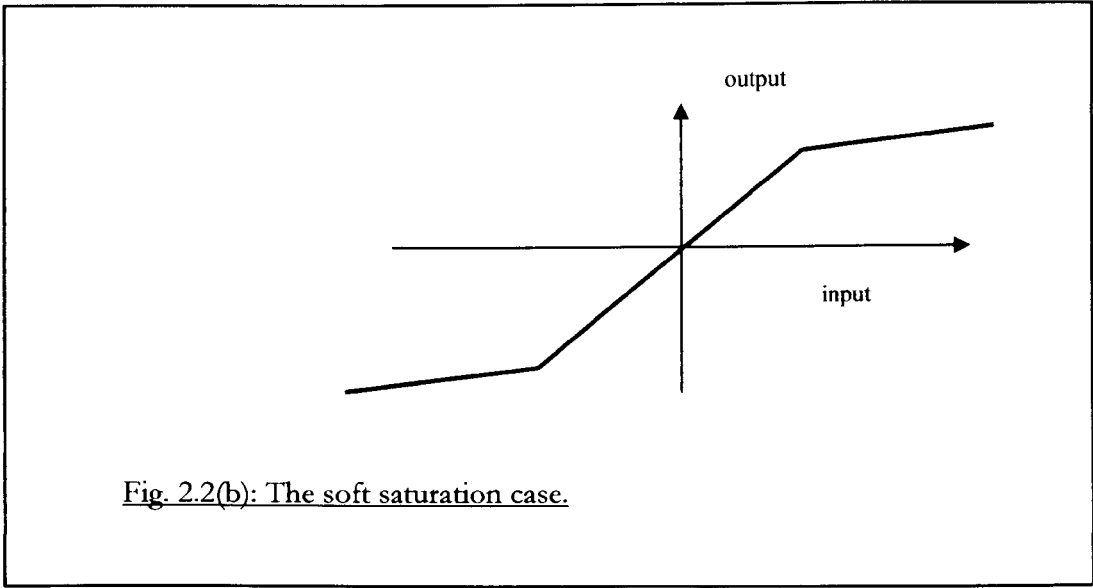
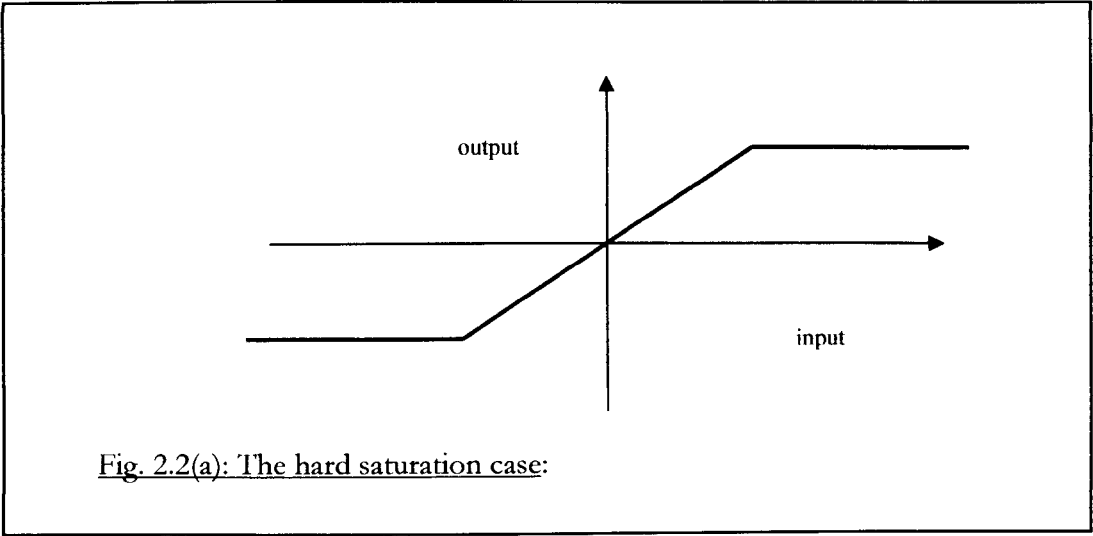
outputs are continuously differentiable with respect to the inputs, although it does not follow that their describing functions are easily integrable, if indeed they are integrable at all. Such non-linearities are quite common and are often deliberately introduced into systems (to stabilize electronic circuits, for example).

2.3.2 Discontinuous non-linearities

For these, the input-output characteristics cannot be modelled by continuous functions over their entire range and they are not continuously differentiable – they contain discontinuities. They can be either single-valued or multi-valued non-linearities.

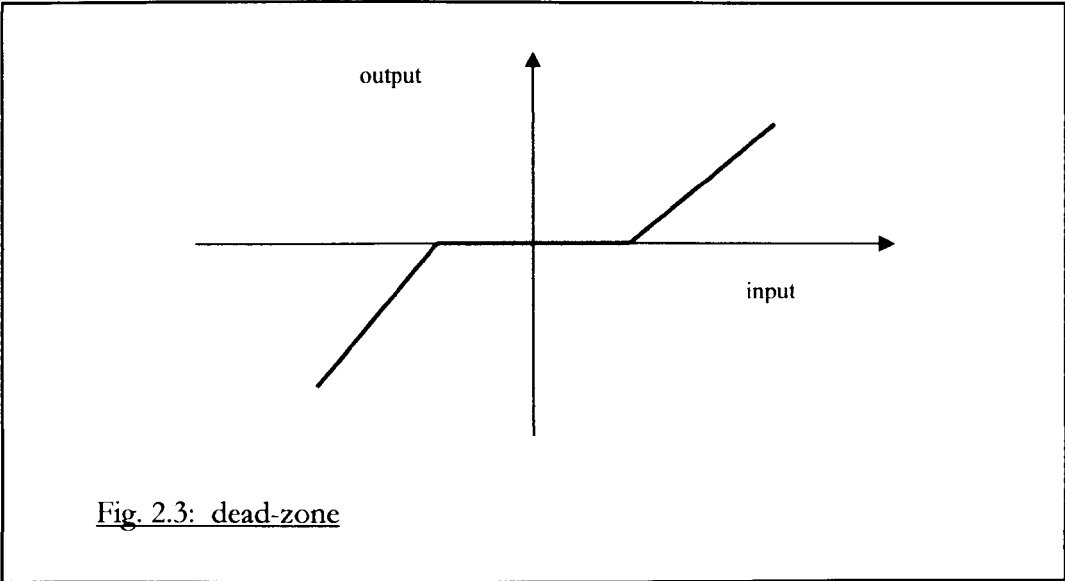
2.3.2.1 Saturation

Every system can exhibit this non-linearity since all physical systems have limits on their range of performance. There are two forms of this non-linearity: hard and soft saturation. Figure 2.2(a) depicts hard saturation; a typical example being when a mechanical valve reaches its end stops. Soft saturation occurs when the slope of the characteristic after the discontinuity, or break point, is not zero but still has a slight slope, at least initially. In practice, in such a case there will usually be a more gentle curved change to the new value rather the abrupt behaviour seen with the hard-spring.



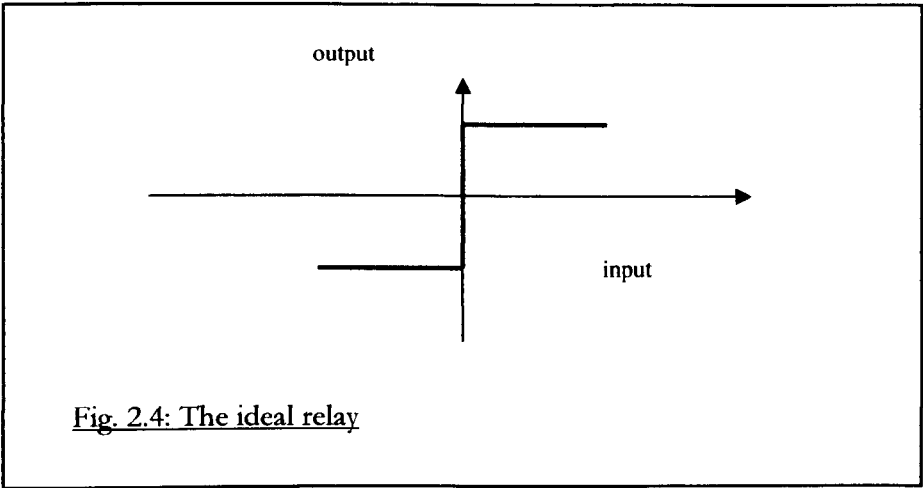
2.3.2.2 Dead zone (or dead-space)

Many systems cannot respond to a small initial input signal but otherwise behave in a linear fashion after the input magnitude has passed a certain minimum value.



2.3.2.3 The ideal relay

The output changes between two distinct states (\pm some voltage level, for example) as the input goes from some negative value, through zero, to a positive value. It is an ideal response to which some non-linearities approach (e.g. an electro-mechanical relay) but most real system are more complicated than this.



2.3.2.4 An ideal relay with dead zone

An example of this is a real electromagnetic relay. A certain amount of current is needed before its coils are able to actuate the device. Hence there is a dead zone as shown in Figure 2.5.

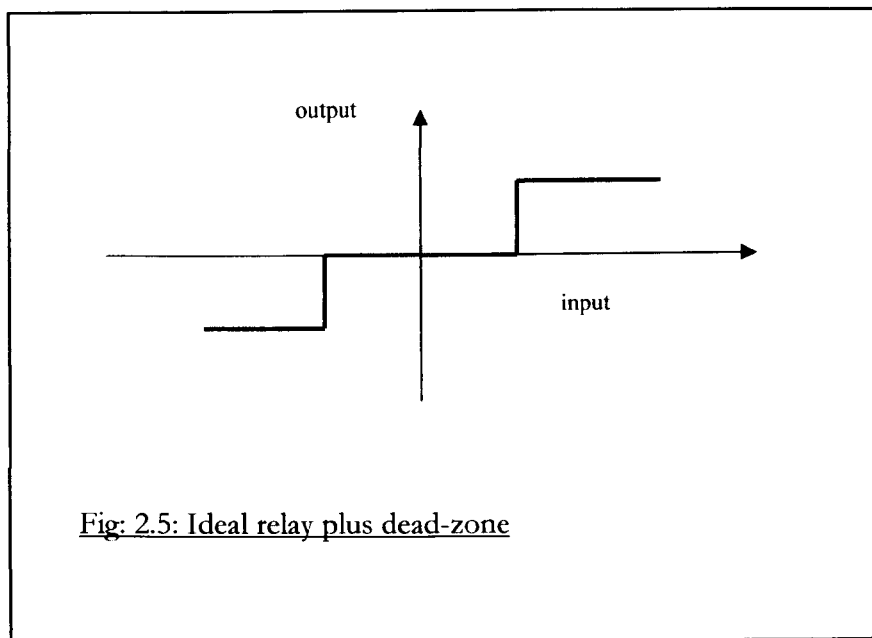
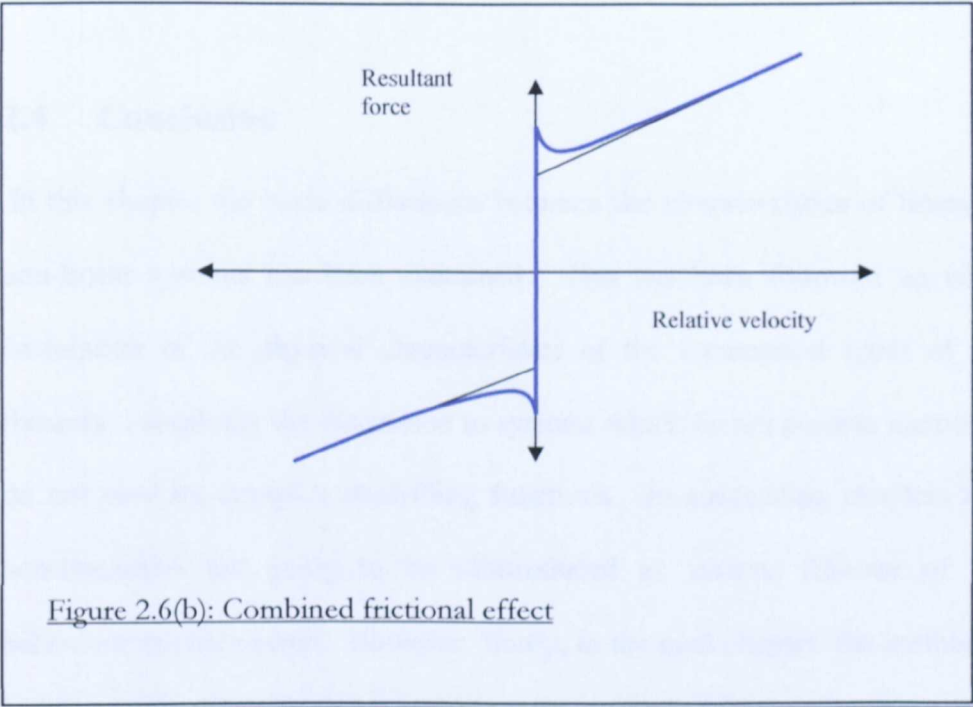
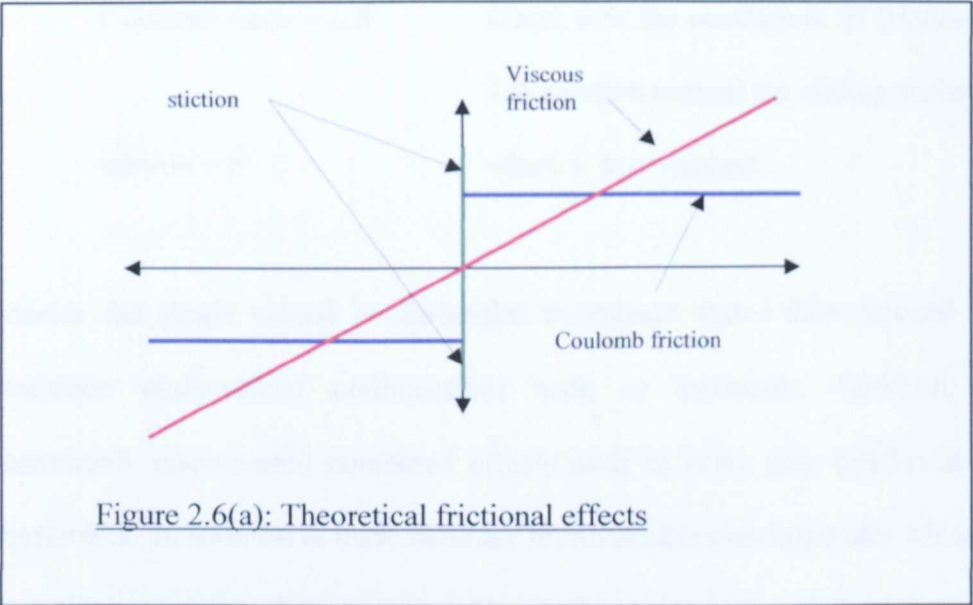


Fig: 2.5: Ideal relay plus dead-zone

2.3.2.5 Friction

This is a force which is encountered wherever there is relative movement between bodies, no matter whether those bodies are solid, liquid or gas. The predominant frictional force is *viscous friction* which is proportional to the relative velocity of the sliding interfaces between the bodies. This viscous force is directly proportional to the relative motion so it is a linear effect. Another dynamic effect is *Coulomb friction*. It is proportional to the normal reaction between those surfaces, the relation between them being called the coefficient of friction. Finally, there is also a static effect, known as *stiction*, which is the additional force required to start the initial relative motion between the bodies. Strictly, this gradually decreases with relative velocity

until it becomes the Coulomb force at relatively low velocities. The theoretically separate effects are shown in Figure 2.6a and the combined practical result is shown in Figure 2.6b.



viscous force = $f \cdot dx/dt$

where f is a constant, and dx is

displacement of one surface with respect to the other.

Coulomb force = $\mu \cdot R$

where μ is the coefficient of friction and R is reaction normal the sliding surfaces.

stiction = F

where F is a constant.

Beside the single valued nonlinearities mentioned above there several very common multi-valued nonlinearities such as hysteresis, backlash, and commonly encountered combined effects such as relay plus dead-zone and hysteresis. In addition to these there are multivariable non-linearities which are functions of more than one variable, such as servo-motors and transistor characteristics.

2.4 Conclusion

In this chapter the basic differences between the characteristics of linear and non-linear systems has been examined. This has been followed up with a description of the physical characteristics of the commonest types of non-linearity – confining the discussion to systems which do not possess memory or do not produce complex describing functions. In succeeding chapters these non-linearities are going to be reintroduced as various features of their behaviour are uncovered. However, firstly, in the next chapter, the methods of analyzing this behaviour have to be discussed and decisions made about which methods are the most appropriate.

Chapter Three

METHODS OF ANALYSING NON-LINEAR SYSTEMS

3.1 Introduction

Several main approaches to the problem of analyzing non-linear behaviour have been developed. The first approach, a time-domain method, was developed by Poincaré (1881, 1882) and used a graphical approach which possesses features which made it potentially useful in this investigation. Another time-domain approach was used by Lyapunov in 1892 (English translation 1961). His techniques were unknown outside of Russia until Lur'e (1957) and LaSalle et al. (1961) brought his work to the attention of the western control engineering community. They have mostly been used to investigate the stability of any system whose model could be expressed in a generalized state-space form. However, his methods do not lend themselves to graphical techniques and do not help in predicting limit-cycle behaviour. Three main frequency-domain techniques have been developed: The describing function technique gradually evolved from the work of several different research groups in the 1940s, (Tustin, 1947; Dutilh, 1950; Oppelt, 1948; Kochenburger, 1950), but was only reported and the approach formalized, principally by Kochenburger, in the 1950s. Two other frequency-domain techniques which have been developed are Popov's (1962) method and Zames' Circle criterion, (Zames *et al.*, 1968; Arcak *et al.*, 2003). However, neither of these used graphical methods to make stability predictions and were

not specifically concerned with limit-cycle prediction. Only two of the techniques, the phase-plane and the describing function methods, provide a graphical approach which is suited to this investigation. These two methods are briefly explained and their advantages and disadvantages compared. The approach chosen for this research is then explained in more detail. An explanation of the ways of using this for stability analysis is then outlined.

In particular, as far as this research is concerned, the main effort has been to investigate graphical methods because, although the chosen approach has limitations, as will be discussed later, other techniques only give information about the range of stable conditions and tend to be rather conservative in their results. Also they do not help in investigating limit-cycle behaviour although the majority of instabilities in non-linear systems occur as limit-cycles, (Atherton, 1981).

3.2 The Phase-Plane Approach

Before the computer age the behaviour of second-order systems was investigated by sketching their phase-plane trajectories. It was the first method (Gibbs, 1928) which aimed to analyse non-linear behaviour by graphical means rather than attempting to linearize differential equations. Unfortunately, before its introduction, if the non-linearity was anything but minor any attempt to linearize it did not remove its effects and consequently progress had been slow. The results which a phase-plane produces are a useful way of depicting non-linear behaviour and the approach can handle extremely non-linear conditions. Also there is a major advantage in that if a limit-cycle is present it will have a

distinct configuration: it will show up as an isolated closed path on the phase-plane portrait. Trajectories near to the limit-cycle will either converge to, or diverge from, but they cannot cross it. The amplitudes of such limit-cycles can be read directly from the phase-plane portrait. Unfortunately the phase-plane approach did not provide an obvious method of design which would fit in with systems of order greater than two and for this reason it was not used in this investigation.

3.3 The Describing Function Approach

This is a frequency domain method which was developed by several groups independently in the 1940s (Atherton, 1975) but publication did not appear until much later. It allows the use of polar plots to investigate the stability conditions of a system. In particular it allows the prediction and characterization of limit cycles.

Compared with the phase-plane approach (Dutton *et al.*, 1997), its strengths are:

- It is applicable to systems whose linear part is of any order. In general the method works better for higher-order systems.
- It works for discontinuous as well as continuous nonlinearities and it has been proved to work for systems which contain more than one discontinuous function.
- It uses simple graphical methods from which both frequency response and amplitude information can be easily obtained.

The approximations inherent in the systems lead to a few limitations:

- It assumes that the plant as a whole acts as a low-pass filter.
- It cannot cope with time-varying systems.
- The non-linear characteristics must be symmetrical about the origin.
- Although the method can cope with systems which do not exhibit odd-symmetry the computational overheads to do so are onerous.
- It works best for systems in which the non-linearities are not too severe, e.g. where the root mean square (rms) value of the output fundamental is within 5% of the total rms output (Atherton, 1975). This is a feature where the phase-plane approach is superior because it can cope with highly non-linear systems although its inability to cope with systems higher than second-order ruled it out for the purposes of this investigation.

3.4 The Describing Function Method in Detail

The basic approach is to assume that the input to a non-linearity is sinusoidal

$$x = X \cdot \sin(\omega t) \quad \dots\dots\dots (3.1)$$

It can be shown (Gibson, 1963) that the output can be represented by

$$y = A_0 + A_1 \sin(\omega t) + B_1 \cos(\omega t) + A_2 \sin(2\omega t) + B_2 \cos(2\omega t) + \dots \quad (3.2)$$

If the nonlinearity is symmetric about the origin, and since most systems behave like low-pass filters which cause the magnitude of higher-frequency components to be attenuated, the output reduces to

$$y = A_1 \sin(\omega t) + B_1 \cos(\omega t) \quad \dots\dots\dots (3.3)$$

where $A_1 = \frac{1}{\pi} \int_0^{2\pi} y \cdot \sin(\omega t) \cdot d(\omega t)$ & $B_1 = \frac{1}{\pi} \int_0^{2\pi} y \cdot \cos(\omega t) \cdot d(\omega t) \quad \dots (3.4)$

The solution to this function has to be found in a piecemeal fashion. However there is quarter-wave symmetry so there are no cosine terms in the harmonics; thus $B_1 = 0$ and the output equation reduces to

$$y = A_1 \sin(\omega t) \dots\dots\dots (3.5)$$

Hence A_1 only need to be calculated and because quarter-wave symmetry is also present, this becomes

$$A_1 = \frac{4}{\pi} \int_0^{\frac{\pi}{2}} y \sin(\omega t) . d(\omega t) \dots\dots\dots (3.6)$$

The function which describes the response of the non-linearity to a sinusoidal input is called the *describing function* of the system and is given by

$$N = \frac{A_1}{X} \dots\dots\dots (3.7)$$

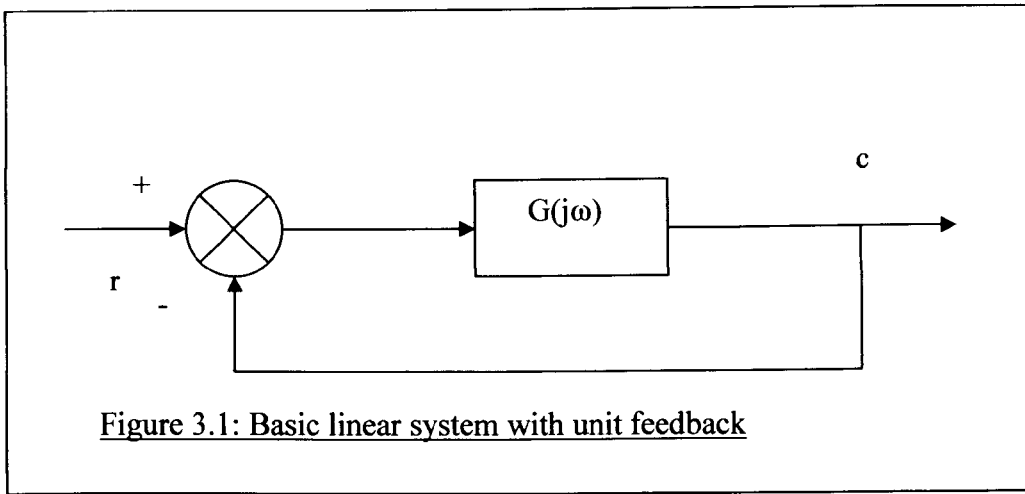
More formally, it is “the ratio of the fundamental output to the magnitude of an applied sinusoidal input” (Atherton, 1996, p 37). However, although the technique was widely applied, the assumption that the method applied to single odd-symmetric and to multiple nonlinear elements was not proven until the late 1950s (Loab, 1956; West and Douce, 1954; Grensted, 1955).

As the concept behind the derivation of a describing function is essentially one of quasi-linearization in which several assumptions are made it might be expected that for more accurate work some harmonic correction terms would be necessary. Johnson (1952) calculated the correction values that should be applied. He found was that the first correction term for the fundamental frequency is zero. Thus the predicted frequency of oscillation should be quite

accurate, which is born out by the results of this investigation, within $\pm 1\%$ of expected values. The accuracy of the predicted amplitudes should be slightly less but again the results between predicted and actual simulated behaviour have been reasonably close, within $\pm 1.5\%$ of expected values.

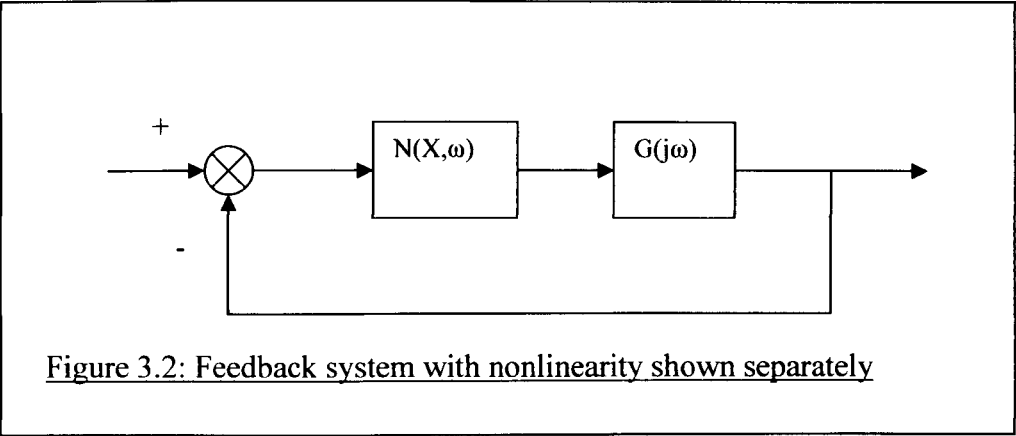
3.5 Stability Analysis using Describing Functions

In a linear system a simple unit feedback arrangement might be as in figure 3.1.



Here the transfer function is $\frac{C(j\omega)}{R(j\omega)} = \frac{G(j\omega)}{1 + G(j\omega)}$ and the corresponding

characteristic equation is $1 + G(j\omega) = 0$. In its simplest open-loop form the Nyquist criterion indicates that the system will be stable whilst $|G(j\omega)| < -1$ at phase crossover (Nyquist, 1932). If the open-loop transfer function $G(j\omega)$ is stable, then the closed loop system is unstable for any encirclement of the point -1. For a non-linear system in which the non-linearity can be replaced by a describing function $N(X, \omega)$, the non-linear function can be placed in series with the linear portion and will obey the usual block-diagram rules (Kochenburger, 1950) as in figure 3.2



The characteristic equation has now become

$$1 + G(j\omega)N(X, \omega) = 0 \quad \dots\dots\dots (3.8)$$

Kochenburger postulated that, because the describing function is effectively a partially linearized description of the non-linearity, the system will be stable whilst

$$|G(j\omega)N(X, \omega)| < -1 \quad \dots\dots\dots (3.9)$$

This extension of the Nyquist criterion is completely heuristic and still has no mathematical basis; however, it works. So, for stability, using a Nyquist diagram with an inverse describing function superimposed the following condition must be satisfied at phase crossover:

$$|G(j, \omega)| < -\frac{1}{|N(X, \omega)|} \quad \dots\dots\dots (3.10)$$

However, it is easier to use the inverse Nyquist diagram, in which case this becomes:

$$\frac{1}{|G(j\omega)|} > -|N(X, \omega)| \quad \dots\dots\dots (3.11)$$

Kochenburger's Stability Criterion:

In order for a system to remain stable, the locus $|G(j\omega)|$ must keep the entire locus $-\frac{1}{|N(X,\omega)|}$ on the right; or the inverse locus $\frac{1}{|G(j\omega)|}$ must keep the locus $-|N(X,\omega)|$ on the left (or must completely enclose the whole of the locus).

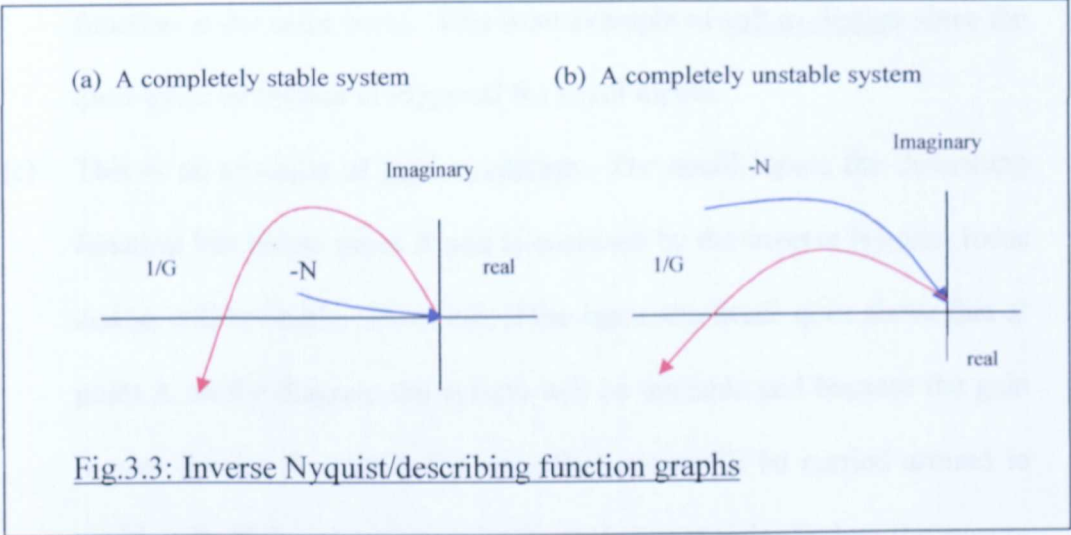
To demonstrate the practical use of Kochenburger's criterion assume that the describing function is a complex quantity, in which case the situations described in Figure 3.3 could exist. This Figure shows some of the consequences of the Kochenburger criterion (the arrows on the describing function loci indicate direction of increasing amplitude whilst the arrows on the inverse Nyquist loci indicate direction of increasing frequency).

The various cases are explained in Figures 3.3 and 3.4. The directions of increasing amplitude of the describing function loci are shown by blue arrows and the directions of increasing frequency of the inverse Nyquist loci are shown by red arrows

Firstly, consider the simplest cases, in Figure 3.3 ((a) and (b)):

- (a) If the describing function locus, in blue, stays entirely to the left of the inverse Nyquist locus, in red, as frequency increases then the system is entirely stable. The gain in this region being less than unity.
- (b) If the describing function locus stays entirely to the right of the inverse Nyquist locus as frequency increases (again as shown by the direction of

the red arrow) then the system is entirely unstable. The gain in this region being greater than unity.



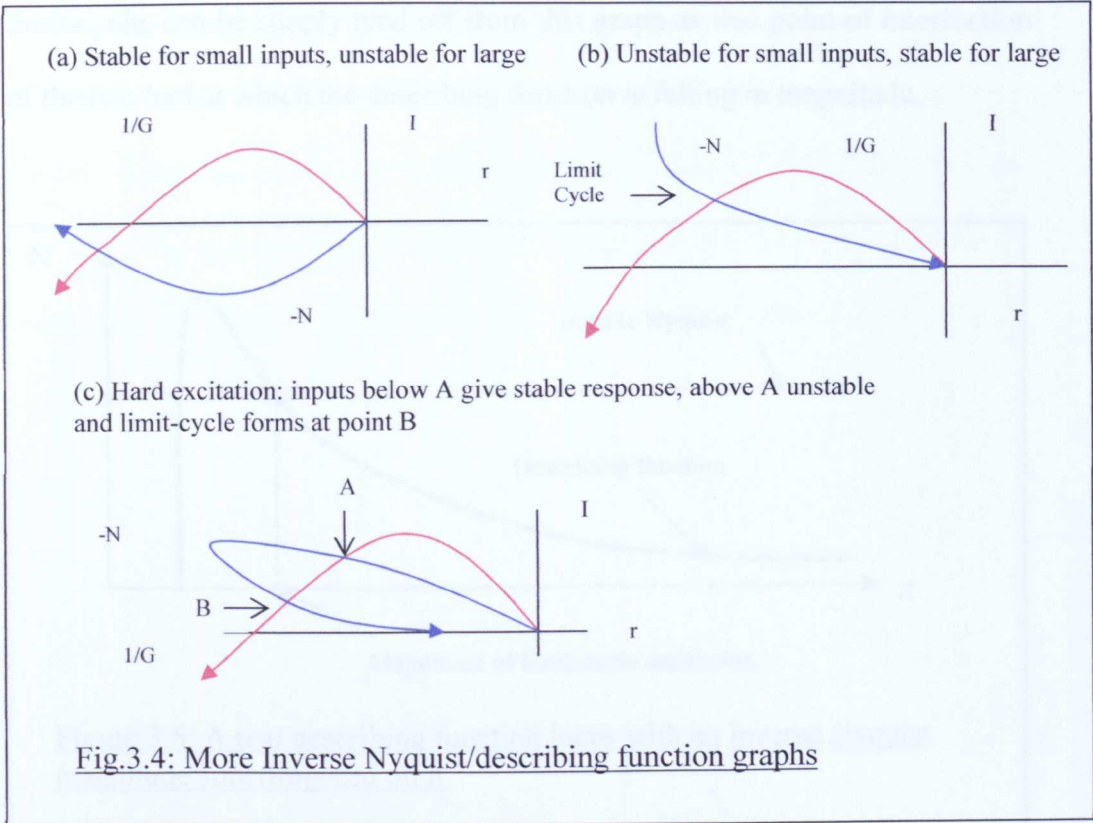
Secondly, Figure 3.4 ((a), (b) and (c)) demonstrates more complicated cases:

- (a) If, for small input magnitudes the describing function locus is to the left of the inverse Nyquist locus (it is enclosed, or to the left of the inverse Nyquist locus in the sense shown by the direction of the red arrow) as the frequency increases, the system is stable but if it is not enclosed for large input amplitudes the system is unstable and will become more so. i.e. the describing function locus moves from a region where the gain is less than unity to region where it is greater than unity
- (b) This case shows a system which is unstable for small amplitudes but stable for large. Small disturbances lie outside the inverse Nyquist locus and grow until the crossover point is reached when the gain becomes unity and the phase difference is 180° . If the amplitude increased beyond this point the gain would fall below unity and the point on the describing function would fall back to the crossover position. Hence there is a stable

oscillation at crossover (a limit-cycle). The frequency of the oscillation will be given by the value of the radial frequency on the inverse Nyquist locus at crossover and the magnitude by the magnitude of the describing function at the same point. This is an example of soft excitation since the limit-cycle behaviour is triggered for small inputs.

- (c) This is an example of hard excitation. For small inputs the describing function lies below point A and is enclosed by the inverse Nyquist locus and so will be stable. However, if the input amplitude goes above that at point A on the diagram the system will be unstable and because the gain is now greater than unity the operating point will be carried around to position B. If the operating point is carried past point B then the system gain becomes less than unity and it will be carried back to this point where a limit-cycle will occur.

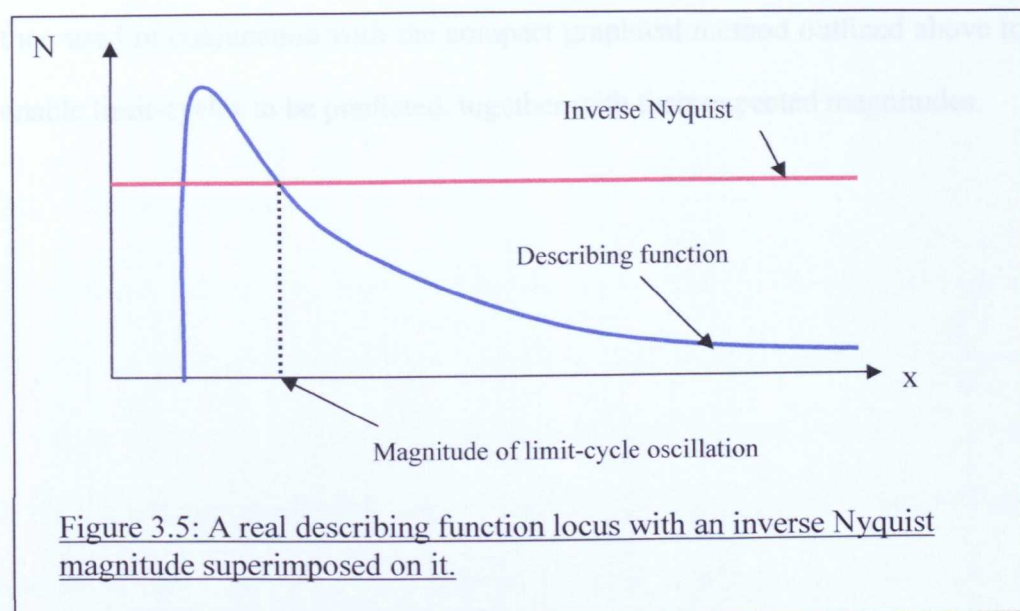
If the describing function had been of the frequency-dependent type then there would be a range of possible describing function loci, but the only one which would matter would be that one which had the same frequency at the intersection with the inverse Nyquist locus as the inverse Nyquist locus itself – all others would decay.



The work reported in this thesis was confined to non-linearities which were not frequency dependent and also did not exhibit hysteresis of any sort. Such non-linearities all have real describing functions. If such describing functions had been used in the examples in Figures 3.3 and 3.4 they would have lain entirely along the real axes and although the same effects would have occurred they would not have been so obvious on the graphs.

However, there is an alternative diagram which can be used in conjunction with the frequency information gained from Figures 3.3 and 3.4: a real describing function can be plotted against the magnitude of the error signal (or the input signal), as in Figure 3.5, and the inverse Nyquist value at intersection (gleaned from Figures 3.3 and 3.4) can be superimposed as a horizontal line on it. The amplitude of the describing function, and hence the amplitude of the

limit-cycle, can be simply read off from this graph as that point of intersection of the two loci at which the describing function is falling in magnitude.



3.6 Conclusion

A survey of the main techniques for stability analysis has been presented in this chapter. The reasons why only two of these techniques were suitable for this investigation was explained and their advantages and disadvantages were summarized and reasons given for deciding to use the describing function technique. The manner in which Nyquist's methods could be combined with describing functions was then presented and the main expected outcomes from the interaction of these two loci were delineated. Finally, because this investigation was dealing with real describing functions a more compact and informative graphical method using the inverse Nyquist locus was introduced.

This more compact method has become the method of choice for subsequent analysis. In the next chapter a general solution is obtained which enables all real describing functions to be easily produced. These describing functions are then used in conjunction with the compact graphical method outlined above to enable limit-cycles to be predicted, together with their expected magnitudes.

Chapter Four

A GENERAL SOLUTION FOR A FAMILY OF REAL DESCRIBING FUNCTIONS

4.1 Introduction

This chapter takes the describing method described in chapter three and presents a graphical method for obtaining describing functions. It then proceeds to use the technique to develop a general solution for obtaining the family of real describing functions. These are the describing functions whose non-linear characteristics are the same irrespective of whether the magnitude of the input signal is increasing or decreasing. Obtaining a general solution also permits the creation of algorithms for its implementation and one such algorithm is presented in section 4.3. Although the general methods of deriving complex describing functions are well known, to the best of the author's knowledge this general solution for generating real describing functions has not been reported in the literature and neither has the associated algorithm.

The purpose of developing a general method is to enable easy and rapid delineation of the describing functions for the non-linearities used in this investigation. Using this result, the describing functions for various common linearities and for a couple of cases, which have been of particular use in this research, are easily obtained. The describing functions for these special cases

are developed in section 4.4. To round off this topic, a method of dealing with continuous functions is explained and concludes with an example.

4.2 The Graphical Method

As described earlier the development of the graphical describing function technique can be traced to several groups working independently (Tustin, 1947; Dutilh, 1950; Oppelt 1948; Kochenburger, 1950). However these wartime developments did not come into general use until the mid 1950s (as reported by Atherton, 1975 and 1997). It is basically an harmonic balance approach modified for feedback control. This meant that only the principal harmonic was used and higher-order oscillations were considered to be negligible due filtering influence of the inertia inherent in the overall process which was being controlled. The basic graphical approach is shown in Figure 4.1. It is assumed that the input is sinusoidal (bottom left-hand corner of the diagram) and this is mapped via the non-linearity (represented at the top left-hand corner of the diagram) to an output. The input has n input magnitude x plotted against time t . The non-linear transformation translates the input magnitude x , on the horizontal axis, to the output magnitude y on the vertical. The output is a plot of output y against time t . This is the same time period as for the input signal and hence the output signal y , due to the non-linear transformation is correlated with the input.

Assuming a sinusoidal input, then the input equation will be given by:

$$x = X \cdot \sin(\omega t) \dots\dots\dots (3.1)$$

and the output equation will be:

$$y = A_1 \sin(\omega t) \dots\dots\dots (3.5)$$

Hence the describing function is

$$N(X, \omega) = \frac{A_1}{X} \dots\dots\dots (3.7)$$

with

$$A_1 = \frac{4}{\pi} \int_0^{\frac{\pi}{2}} y \cdot \sin(\omega t) \cdot d(\omega t) \dots\dots\dots (3.6)$$

The method is a quasi-linearization process in which a section of a static non-linearity is represented by a gain which depends on the magnitude of the input signal (Atherton, 1981; Gibson, 1963). For this reason it is assumed that the non-linearity consists of lines of constant slope to each side of the break-points, the integrations can be calculated in a piecewise fashion to give:

$$A_1 = \frac{4}{\pi} \left[\int_0^{\alpha_1} y \cdot \sin(\omega t) \cdot d(\omega t) + \int_{\alpha_1}^{\alpha_2} y \cdot \sin(\omega t) \cdot d(\omega t) + \int_{\alpha_2}^{\frac{\pi}{2}} y \cdot \sin(\omega t) \cdot d(\omega t) \right] \dots\dots (4.1)$$

where the individual values of y in each of the separate linear sections has the form

$$y = Kx + c \dots\dots\dots (4.2)$$

in which K is the slope of the relevant section. The positions at which the slopes K abruptly change value have been termed **break points** in this investigation.

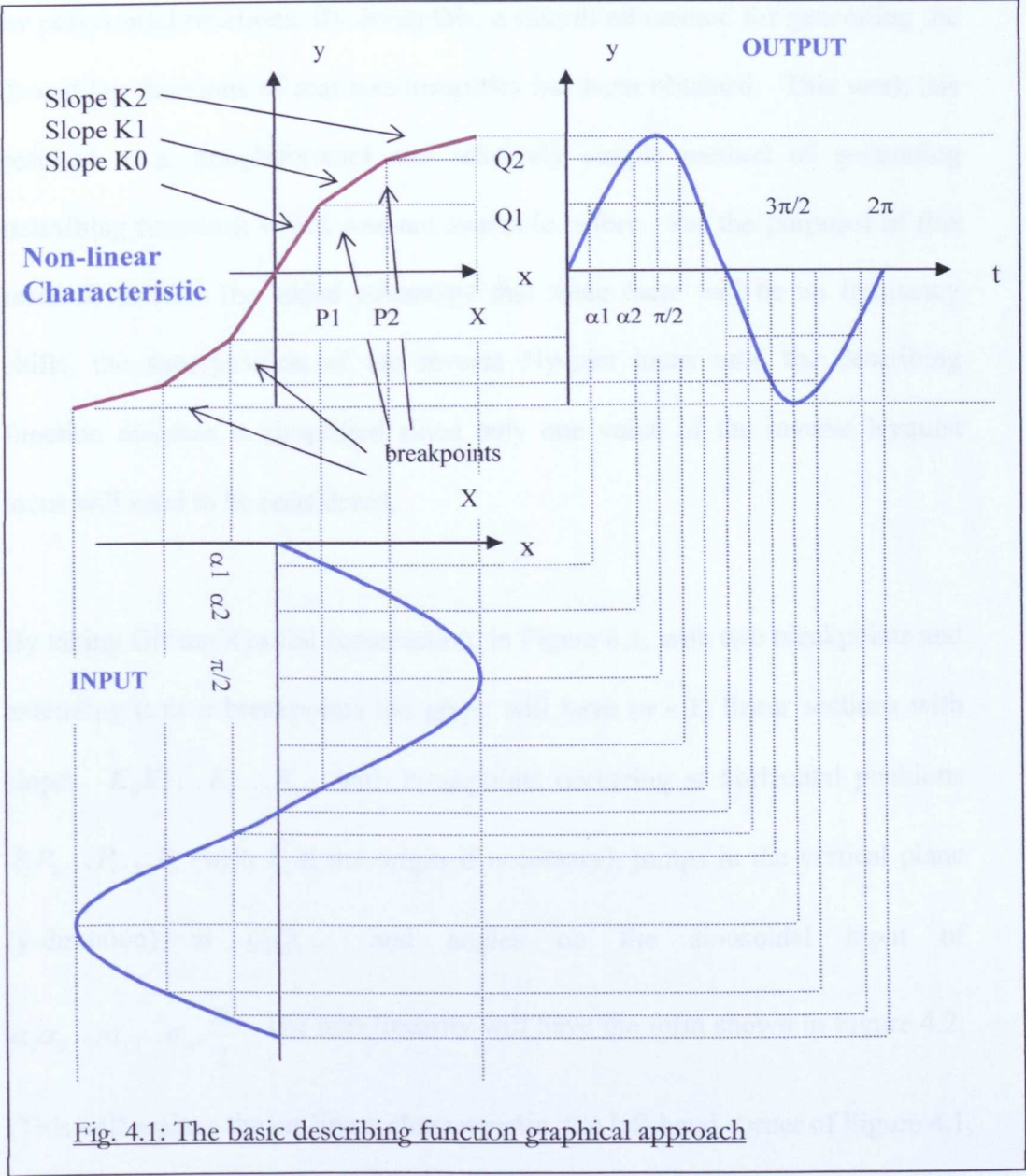


Fig. 4.1: The basic describing function graphical approach

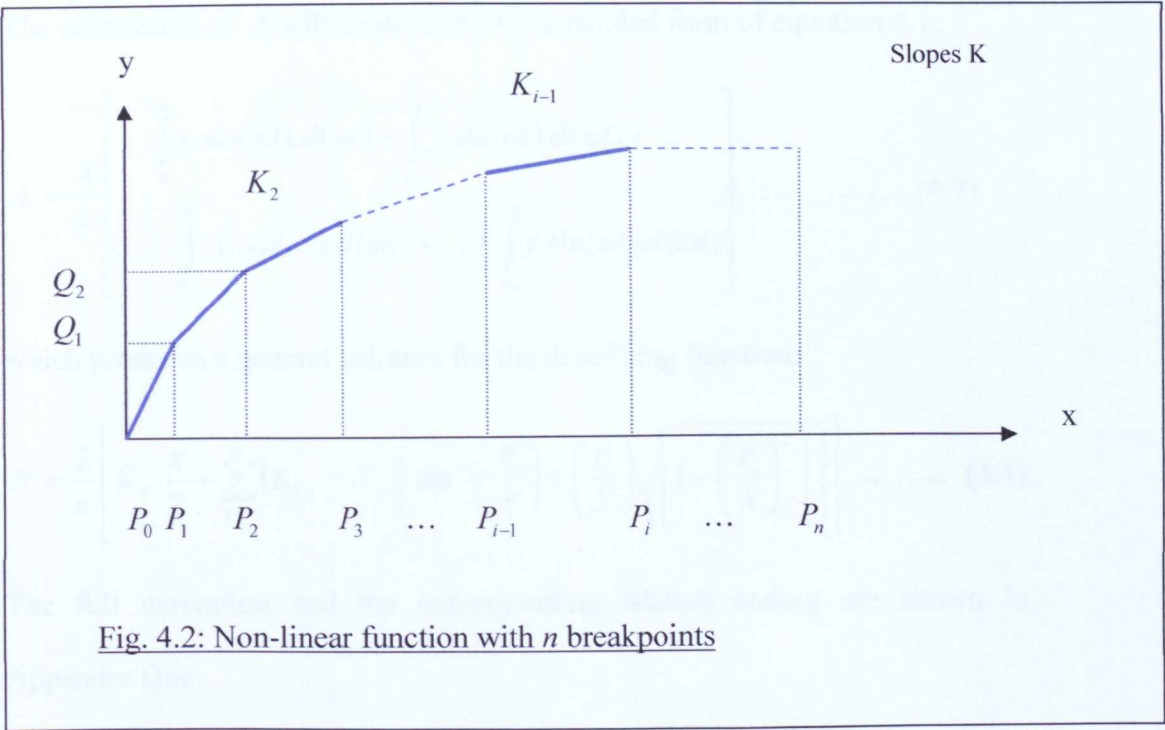
4.3 A General Solution

Gibson (1963) showed how to obtain a general describing function by using this piecewise linear approach but aimed for an overall solution. It is the author’s opinion that various features of this general method can be made clearer by considering single-valued non-linearities separately from double, or multi-valued, non-linearities. Single-valued non-linearities will produce real, as opposed to complex, describing functions and they can often be described

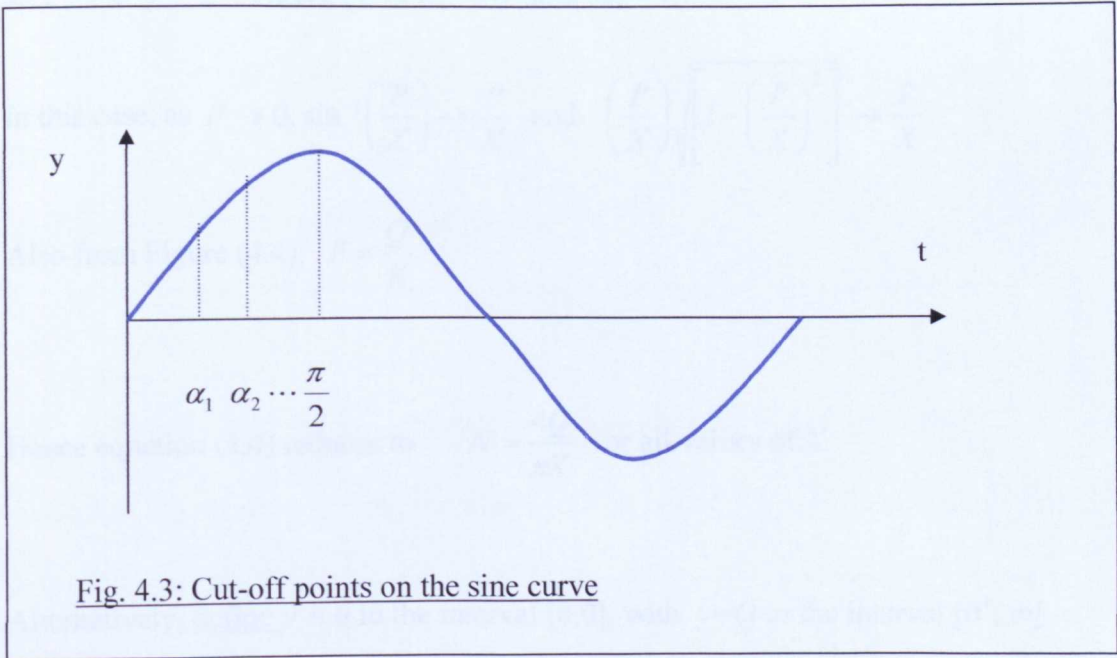
by polynomial functions. By doing this, a simplified method for generating the describing functions of real non-linearities has been obtained. This work has resulted in a straightforward and relatively simple method of generating describing functions which was not available before. For the purposes of this research there is the added advantage that since there will be no frequency shifts, the superposition of the inverse Nyquist locus onto the describing function diagram is simplified since only one value of the inverse Nyquist locus will need to be considered.

By taking Gibson's initial construction, in Figure 4.1, with two breakpoints and extending it to n breakpoints the graph will have $(n - 1)$ linear sections with slopes $K_0 K_1 \dots K_i \dots K_n$, with breakpoints occurring at horizontal positions $P_1 P_2 \dots P_i \dots P_n$ (with P_0 at the origin if necessary), jumps in the vertical plane (y-direction) at $Q_1 Q_2 \dots$ and angles on the sinusoidal input of $\alpha_1 \alpha_2 \dots \alpha_i \dots \alpha_n, \frac{\pi}{2}$. The non-linearity will have the form shown in Figure 4.2.

(This will replace the on-linear characteristic, top left-hand corner of Figure 4.1, which maps the input signal to the output signal)



The cut-off points on the sine curve will occur as in Figure 4.3. (This replaces the sinusoidal-type of output in the top right-hand corner of Figure 4.1.)



The calculation of A_1 will be given by the extended form of equation 4.1:

$$A_1 = \frac{4}{\pi} \left[\begin{aligned} &\int_0^{\alpha_1} y.\sin(\omega t).d(\omega t) + \int_{\alpha_1}^{\alpha_2} y.\sin(\omega t).d(\omega t) + \dots \\ &\dots + \int_{\alpha_{i-1}}^{\alpha_i} y.\sin(\omega t).d(\omega t) + \dots + \int_{\alpha_n}^{\pi/2} y.\sin(\omega t).d(\omega t) \end{aligned} \right] \dots\dots\dots (4.3)$$

which produces a general solution for the describing function:

$$N = \frac{2}{\pi} \left[K_n \cdot \frac{\pi}{2} + \sum_{i=1}^n (K_{i-1} - K_i) \left(\sin^{-1} \left(\frac{P_i}{X} \right) + \left(\frac{P_i}{X} \right) \sqrt{1 - \left(\frac{P_i}{X} \right)^2} \right) \right] \dots\dots\dots (4.4)$$

The full derivation and the corresponding Matlab coding are shown in Appendix One.

If Coulomb friction, or relay action, is present at initial amplitudes then equation (4.4) has to be adjusted.

Consider the case where only Coulomb friction is present:

as $P \rightarrow 0$ $y = Q$, Figure (4.4), $K_0 = \infty$ and $K_1 = 0$.

In this case, as $P \rightarrow 0$, $\sin^{-1} \left(\frac{P}{X} \right) \rightarrow \frac{P}{X}$ and $\left(\frac{P}{X} \right) \sqrt{1 - \left(\frac{P}{X} \right)^2} \rightarrow \frac{P}{X}$

Also from Figure (4.4), $P = \frac{Q}{K}$

Hence equation (4.4) reduces to $N = \frac{4Q}{\pi X}$ for all values of X .

Alternatively, define $y = 0$ in the interval $[0,0]$, with $y=Q$ in the interval $[0^+, \infty]$ and $y = -Q$ in the interval $[-\infty, 0^-]$. The function is now symmetric so

The Algorithm

$$y = 0 + A_1 \cdot \sin(\omega t)$$

The describing function is then given by

$$N = \frac{A_1}{X} = \frac{1}{X} \cdot \frac{4}{\pi} \int_0^{\frac{\pi}{2}} y \cdot \sin(\omega t) \cdot d(\omega t) \text{ and this is integrable.}$$

Since Q is the first term of the describing function, where Q is

$$\therefore N = \frac{4}{\pi X} \int_0^{\frac{\pi}{2}} Q \sin(\omega t) \cdot d(\omega t) = \frac{4Q}{\pi X} \left[\cos 0 - \cos \frac{\pi}{2} \right] = \frac{4Q}{\pi X}$$

where X is the dead-zone break-point.

Stage Three: (a) If saturation is not present make K , the first term of the describing function (K is the gain of the last stage of the non-linearity) or add it to the result of stage one.
(b) If saturation is present then omit this term.

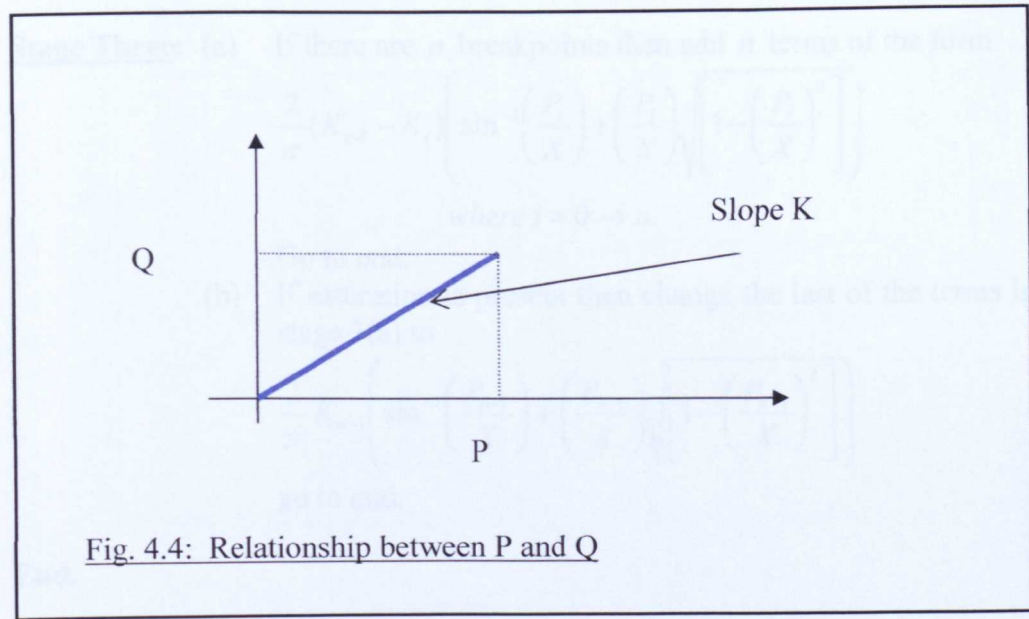


Fig. 4.4: Relationship between P and Q

4.4 An Algorithm for generating real describing functions

A method for using the general solution given in equation (4.4) to generate particular real describing functions which can deal with discrete cases is now presented.

The Algorithm:

If Coulomb friction or relay action is present start at stage one, otherwise start at stage two.

- Stage One:**
- (a) If Coulomb friction or relay action is present then make $\frac{4Q}{\pi X}$ the first term of the describing function, where Q is the value of the Coulomb friction term.
 - (b) If dead-zone is also present multiply the above result by $\sqrt{1 - \left(\frac{P}{X}\right)^2}$, where P is the dead-zone break-point.

- Stage Two:**
- (a) If saturation is not present make K_n the first term of the describing function. (K_n is the gain of the last stage of the non-linearity.) or add it to the result of stage one.
 - (b) If saturation is present then omit this term.

- Stage Three:**
- (a) If there are n breakpoints then add n terms of the form

$$\frac{2}{\pi} (K_{i-1} - K_i) \left(\sin^{-1} \left(\frac{P_i}{X} \right) + \left(\frac{P_i}{X} \right) \sqrt{1 - \left(\frac{P_i}{X} \right)^2} \right)$$

where $i = 0 \rightarrow n$.

Go to end.

- (b) If saturation is present then change the last of the terms in stage 3(a) to

$$\frac{2}{\pi} K_{n-1} \left(\sin^{-1} \left(\frac{P_{n-1}}{X} \right) + \left(\frac{P_{n-1}}{X} \right) \sqrt{1 - \left(\frac{P_{n-1}}{X} \right)^2} \right)$$

go to end.

End.

4.5 Some Typical Examples

Dutton *et al.* (1997, p705) lists some common describing functions. These are easily obtained from the general solution given in equation (4.4). Below are the derivations relevant to this research.

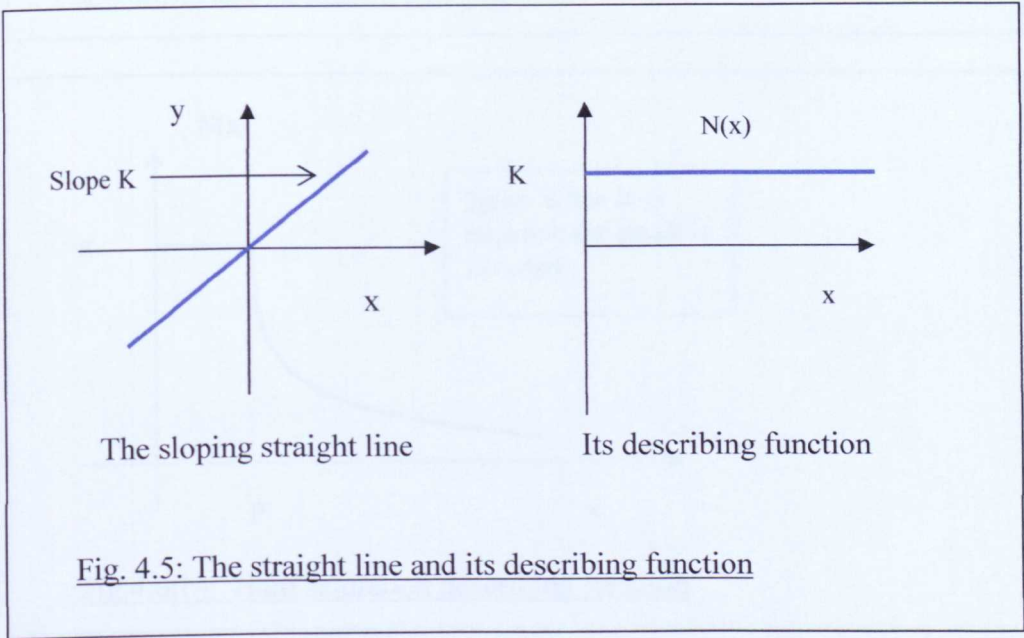
4.5.1 The trivial case: the straight line

The trivial case of a simple straight with no non-linearities is presented for two reasons: (i) it is a useful first test to check that the final describing function equation has the correct form, (ii) the result is needed later in the investigation. There are no break-points so $n = 0$, & $K_i = K$ since there is only one slope.

Equation (4.4), the general equation, reduces to

$$N = \frac{2}{\pi} \left[K \cdot \frac{\pi}{2} + (K - K)(\dots) \right] = K \dots\dots\dots (4.5)$$

This is the slope of a straight line - as expected.



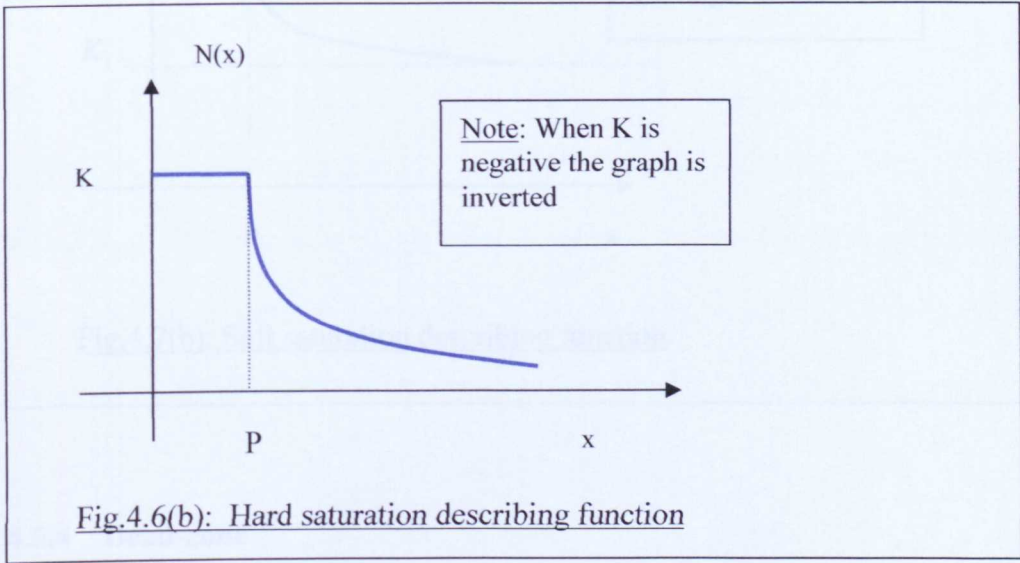
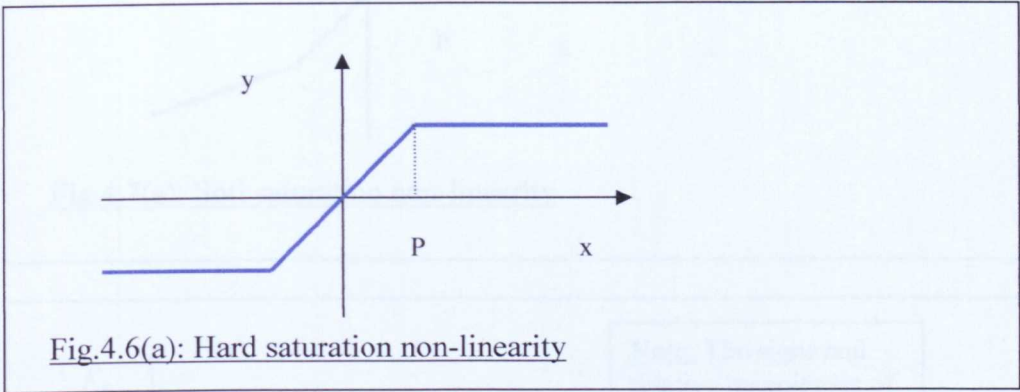
4.5.2 Hard Saturation

This is the classical case shown in all textbooks and papers described simply as ‘saturation’. In practice the input/output characteristic lies somewhere between

the hard and soft saturation cases. With reference to equation 4.4,
 $n = 1, \quad K_0 = K, \quad K_1 = 0, \quad P_i = P$ so the describing function is:

$$N = \frac{2K}{\pi} \left[\sin^{-1} \left(\frac{P}{X} \right) + \frac{P}{X} \sqrt{1 - \left(\frac{P}{X} \right)^2} \right] \dots\dots\dots (4.6)$$

when $X > P$ and $N = K$ otherwise.



4.5.3 Soft Saturation

This is where the slope of the graph above the break point is not zero. As there is only one breakpoint and the slope after it is not zero it follows that:

$n = 0, \quad K_0 = K_0, \quad K_1 = K_1 = K_n, \quad P_i = P,$ so the describing function is

$$N = \frac{2}{\pi} \left[K_1 \cdot \frac{\pi}{2} + (K_0 - K_1) \left[\sin^{-1} \left(\frac{P}{X} \right) + \frac{P}{X} \sqrt{1 - \left(\frac{P}{X} \right)^2} \right] \right] \dots\dots\dots (4.7)$$

when $X > P$ and $N = K_0$ otherwise.

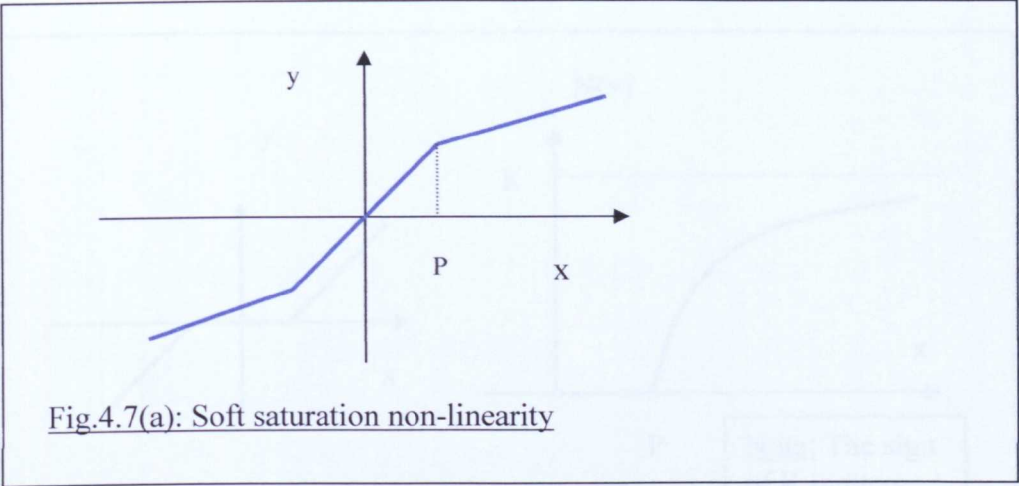


Fig.4.7(a): Soft saturation non-linearity

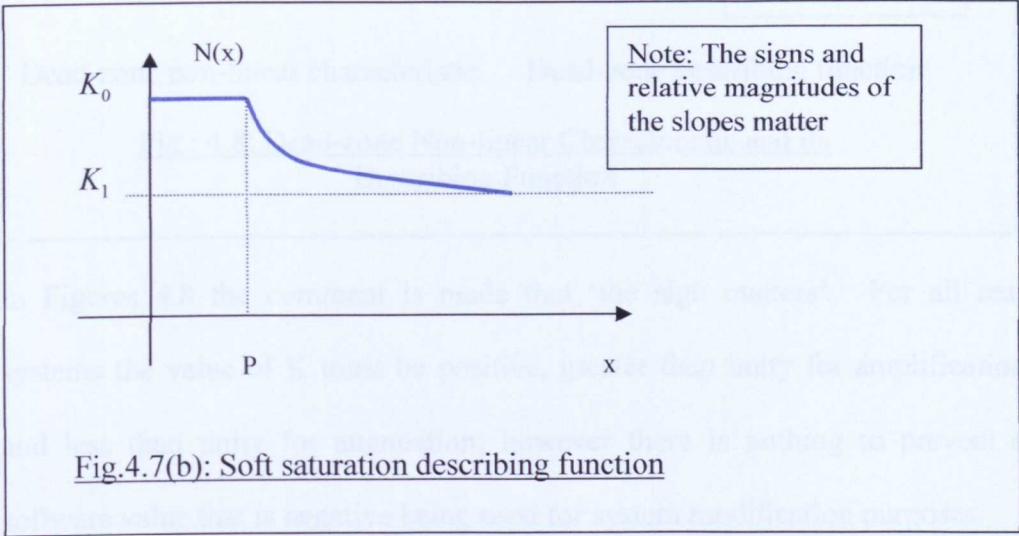


Fig.4.7(b): Soft saturation describing function

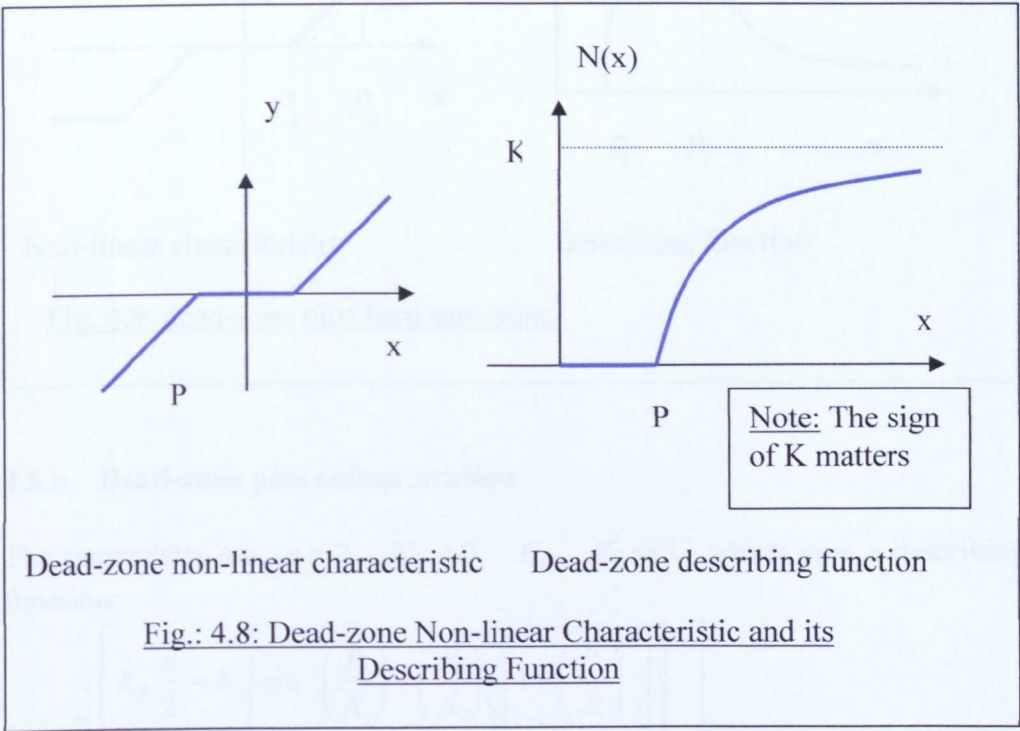
4.5.4 Dead-zone

This is the case in which the process does not react until the input signal is above a certain threshold level, there is effectively one break-point.

$n = 1, \quad K_0 = 0, \quad K_1 = K_n = K, \quad P_1 = P$, giving a describing function

$$N = \frac{2K}{\pi} \left[\frac{\pi}{2} - \sin^{-1} \left(\frac{P}{X} \right) - \frac{P}{X} \sqrt{1 - \left(\frac{P}{X} \right)^2} \right] \dots\dots\dots (4.8)$$

when $X > P$ and $N = 0$ otherwise.



In Figures 4.8 the comment is made that ‘the sign matters’. For all real systems the value of K must be positive, greater than unity for amplification and less than unity for attenuation; however there is nothing to prevent a software value that is negative being used for system modification purposes.

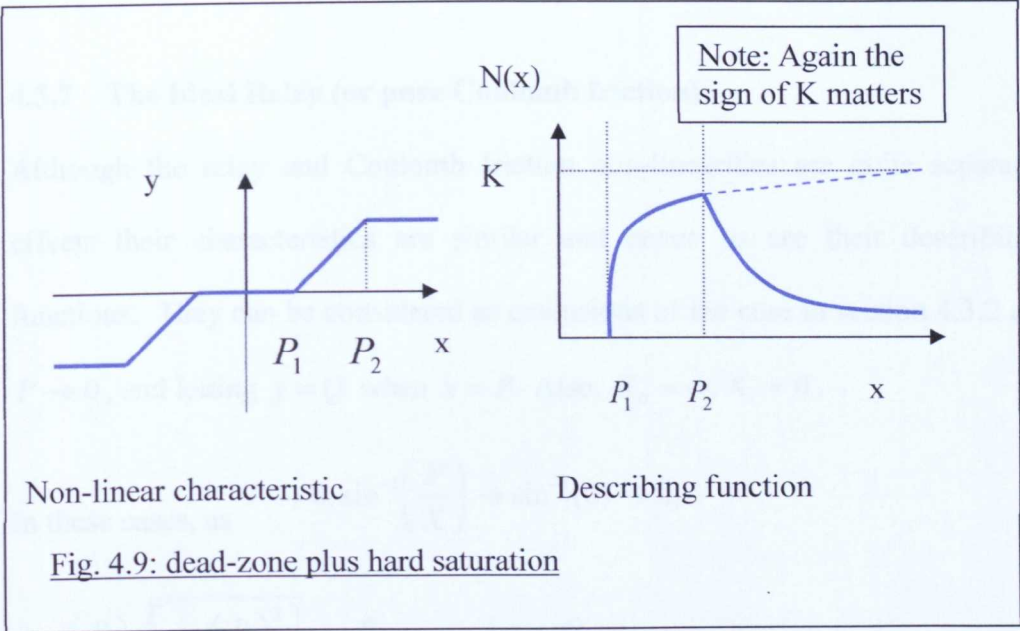
4.5.5 Dead-zone plus hard saturation

$$n = 2, \quad K_0 = 0, \quad K_1 = K, \quad K_2 = 0$$

So the describing function is

$$N = \frac{2K}{\pi} \left[\sin^{-1} \left(\frac{P_2}{X} \right) - \sin^{-1} \left(\frac{P_1}{X} \right) + \left(\frac{P_2}{X} \right) \sqrt{1 - \left(\frac{P_2}{X} \right)^2} - \left(\frac{P_1}{X} \right) \sqrt{1 - \left(\frac{P_1}{X} \right)^2} \right]$$

when $X > P_2$, N as for deadzone when $P_1 < X \leq P_2$ and $N = 0$ when $X \leq P_1$.
..... (4.9)

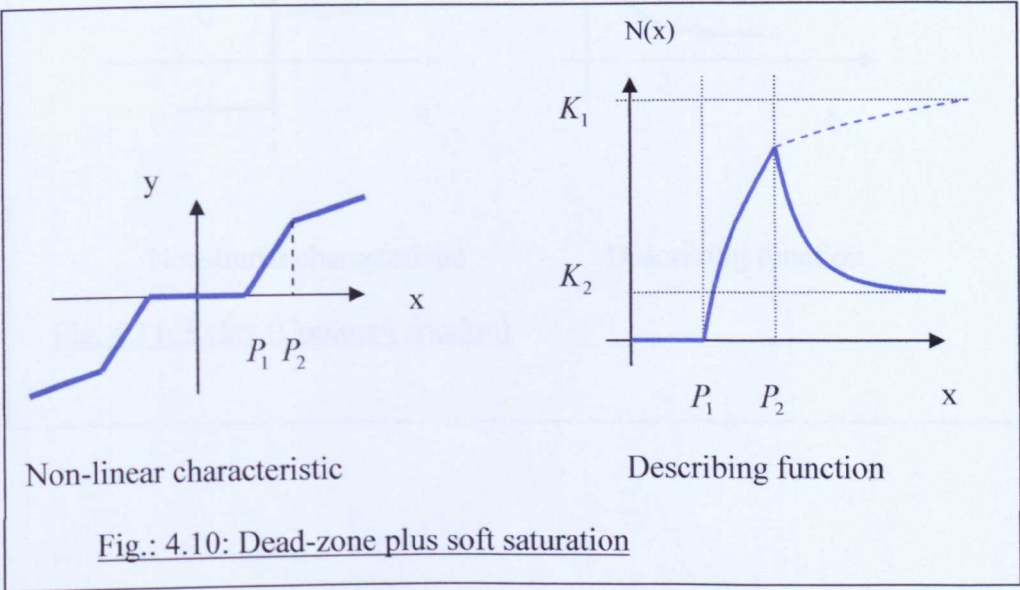


4.5.6 Dead-zone plus soft saturation

The parameters are $n = 2$, $K_0 = 0$, K_1 , $K_2 \ll 1$ which give a describing function:

$$N = \frac{2}{\pi} \left[\begin{aligned} &K_2 \frac{\pi}{2} - K_1 \left[\sin^{-1} \left(\frac{P_1}{X} \right) + \left(\frac{P_1}{X} \right) \sqrt{1 - \left(\frac{P_1}{X} \right)^2} \right] \\ &+ (K_1 - K_2) \left[\sin^{-1} \left(\frac{P_2}{X} \right) + \left(\frac{P_2}{X} \right) \sqrt{1 - \left(\frac{P_2}{X} \right)^2} \right] \end{aligned} \right]$$

when $X > P_2$, n as for deadzone when $P_1 < X \leq P_2$ (4.10)
and $N = 0$ when $X \leq P_1$.



4.5.7 The Ideal Relay (or pure Coulomb friction)

Although the relay and Coulomb friction non-linearities are quite separate effects their characteristics are similar and hence so are their describing functions. They can be considered as extensions of the case in section 4.3.2 as $P \rightarrow 0$, and letting $y = Q$ when $x = P$. Also, $K_0 = \infty$, $K_1 = 0$.

In these cases, as $P \rightarrow 0, \sin^{-1}\left(\frac{P}{X}\right) \rightarrow \sin^{-1}(0) \rightarrow 0,$

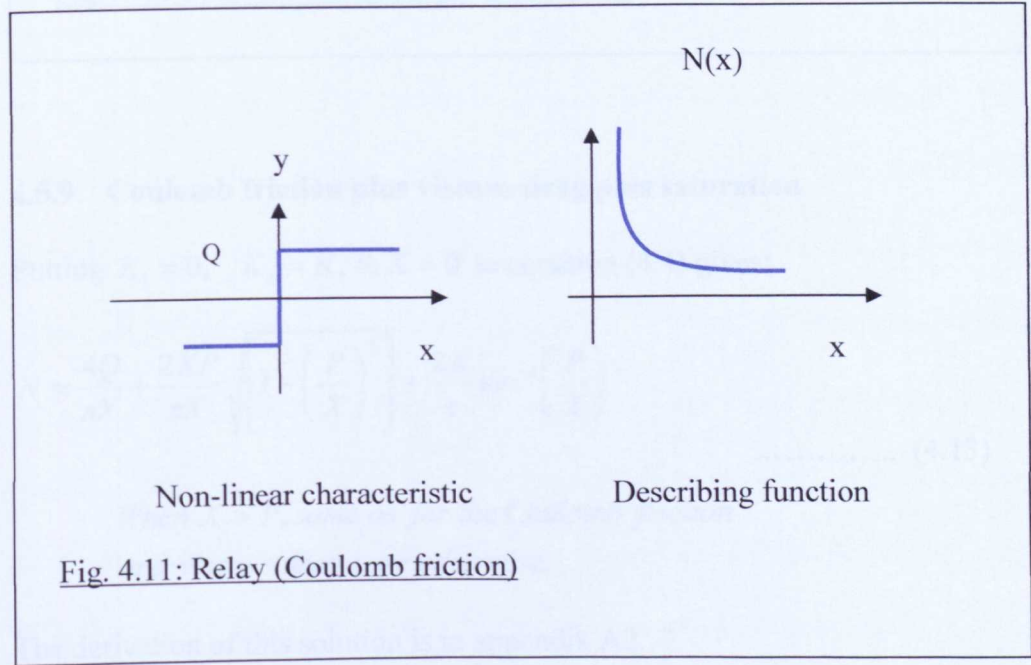
So $\left(\frac{P}{X}\right)\sqrt{1-\left(\frac{P}{X}\right)^2} \rightarrow \frac{P}{X}$ and $K = \frac{Q}{P}$

as demonstrated in Appendix A1.1 stated in the algorithm in section 4.3.

Hence equation (4.4) reduces to

$$N = \frac{4Q}{\pi X} \dots\dots\dots (4.11)$$

for all values of X .



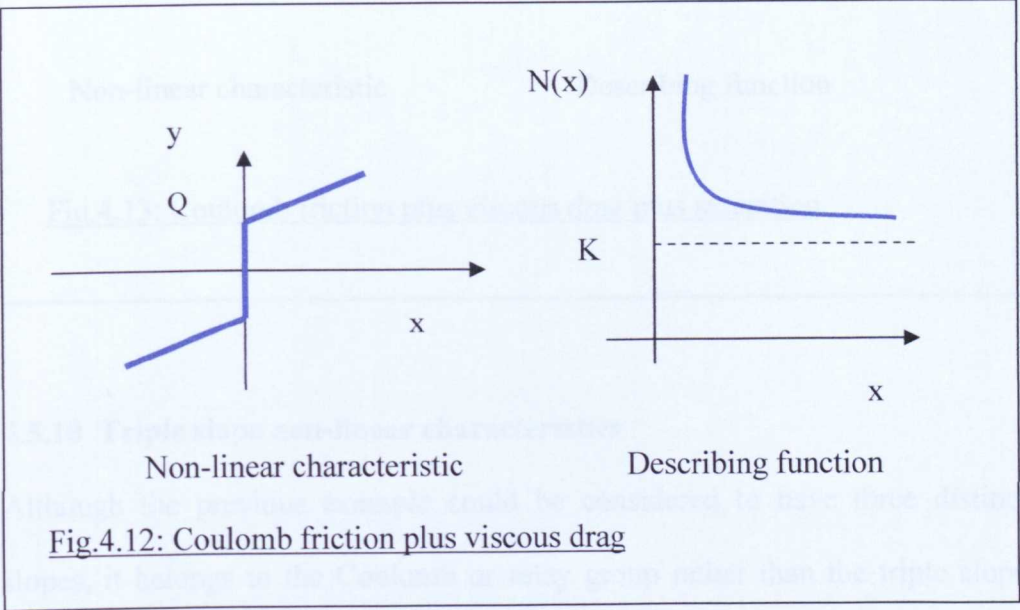
4.5.8 Coulomb friction plus viscous drag

Using similar arguments to the previous case, when $K_0 = \infty$, $K_1 = K$, equation (4.4) becomes:

$$N = \frac{4Q}{\pi X} + K$$

again, for all values of X .

..... (4.12)



4.5.9 Coulomb friction plus viscous drag plus saturation

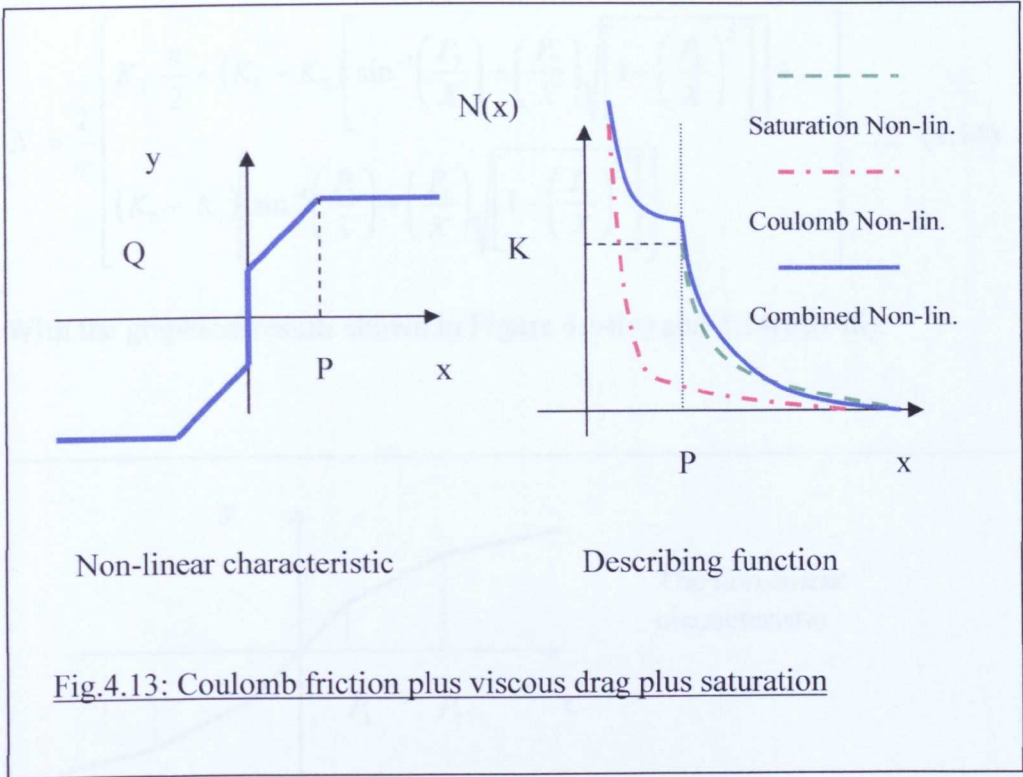
Putting $K_1 = 0$, $K_2 = K$, & $K = 0$ in equation (4.4) gives:

$$N = \frac{4Q}{\pi X} + \frac{2KP}{\pi X} \sqrt{1 - \left(\frac{P}{X}\right)^2} + \frac{2K}{\pi} \sin^{-1}\left(\frac{P}{X}\right)$$

When $X > P$, same as for the Coulomb friction and viscous drag case otherwise.

..... (4.13)

The derivation of this solution is in appendix A2.



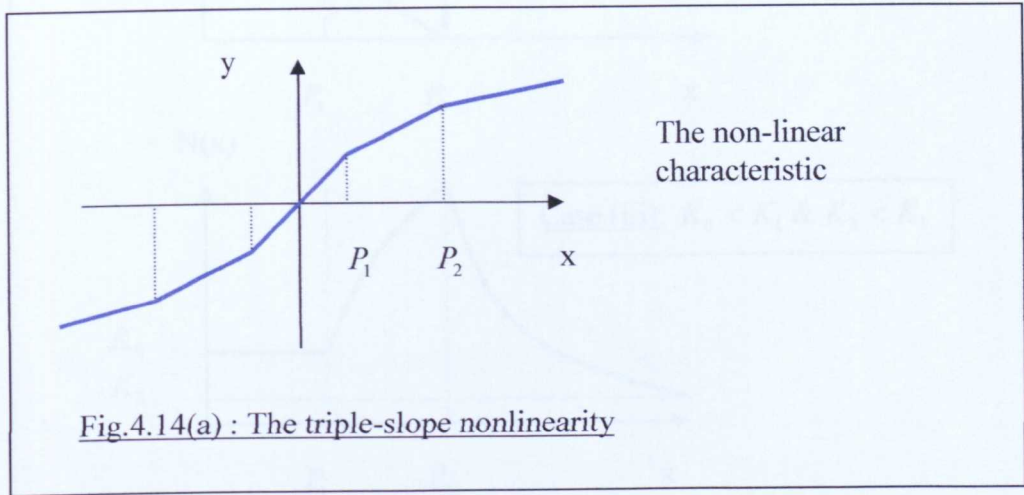
4.5.10 Triple slope non-linear characteristics

Although the previous example could be considered to have three distinct slopes, it belongs to the Coulomb or relay group rather than the triple slope group since its describing function characteristics are quite different. The behaviour of non-linearities with up to three break points was investigated since it was noticed that by judicious choice of the relative values, and also the signs, of the slopes K_0, K_1, K_2 and K_3 of the straight sections, some multi-peak describing functions could be obtained which gave the possibility of multiple crossings of the inverse Nyquist locus. As only real describing functions are being investigated all of the frequencies would be the same but different amplitudes of the limit-cycle oscillations can be obtained.

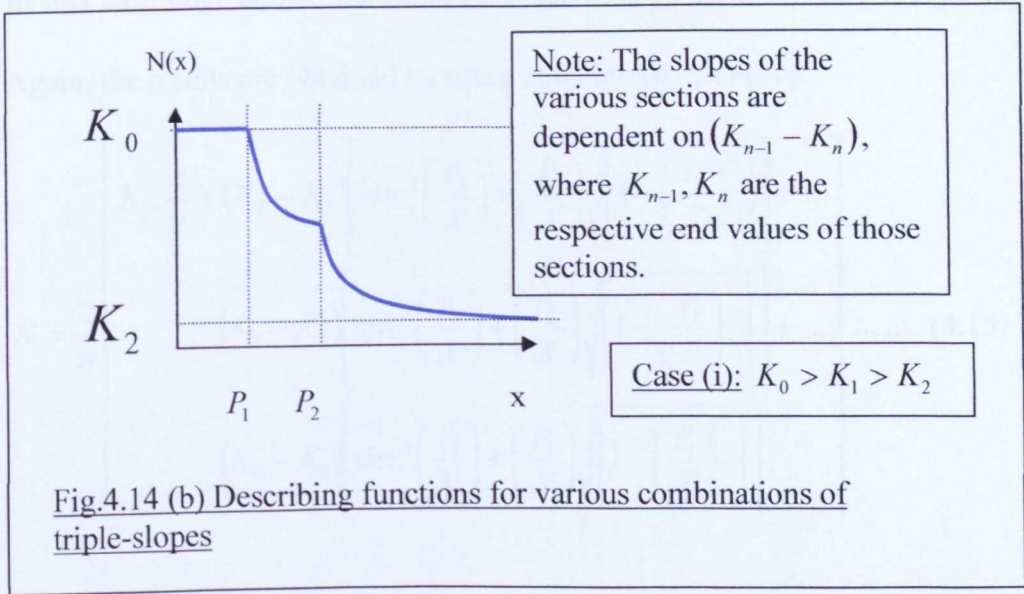
Using equation (4.4) with slopes K_0, K_1, K_2 and breaks at P_1, P_2 gives the equation:

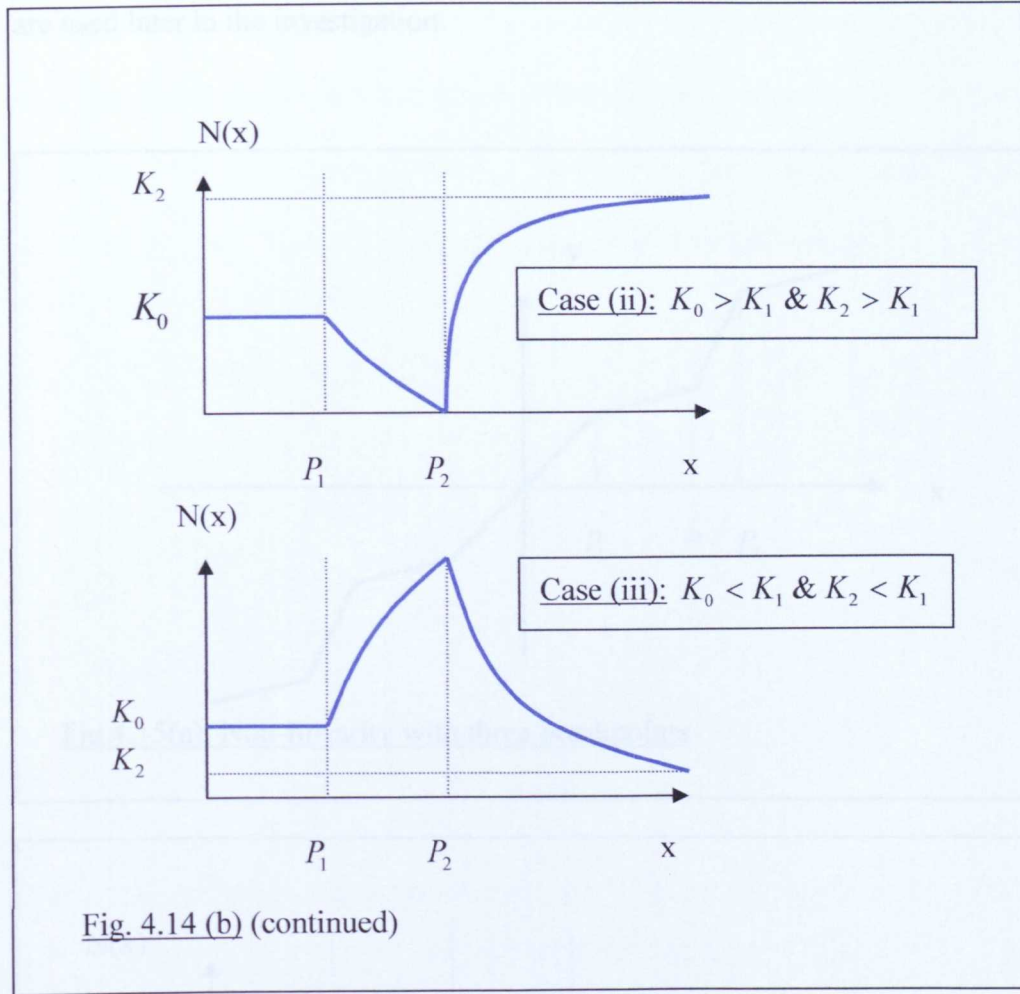
$$N = \frac{2}{\pi} \left[\begin{aligned} &K_2 \cdot \frac{\pi}{2} + (K_1 - K_2) \left[\sin^{-1} \left(\frac{P_2}{X} \right) + \left(\frac{P_2}{X} \right) \sqrt{1 - \left(\frac{P_2}{X} \right)^2} \right] + \dots \\ &(K_0 - K_1) \left[\sin^{-1} \left(\frac{P_1}{X} \right) + \left(\frac{P_1}{X} \right) \sqrt{1 - \left(\frac{P_1}{X} \right)^2} \right] \end{aligned} \right] \dots\dots (4.14)$$

With the graphical results shown in Figure 4.14(a) and 4.14(b)(i-iii).



For this particular non-linearity the shapes of the describing function for several different combinations of the slopes K_0, K_1 & K_2 are important for later work.





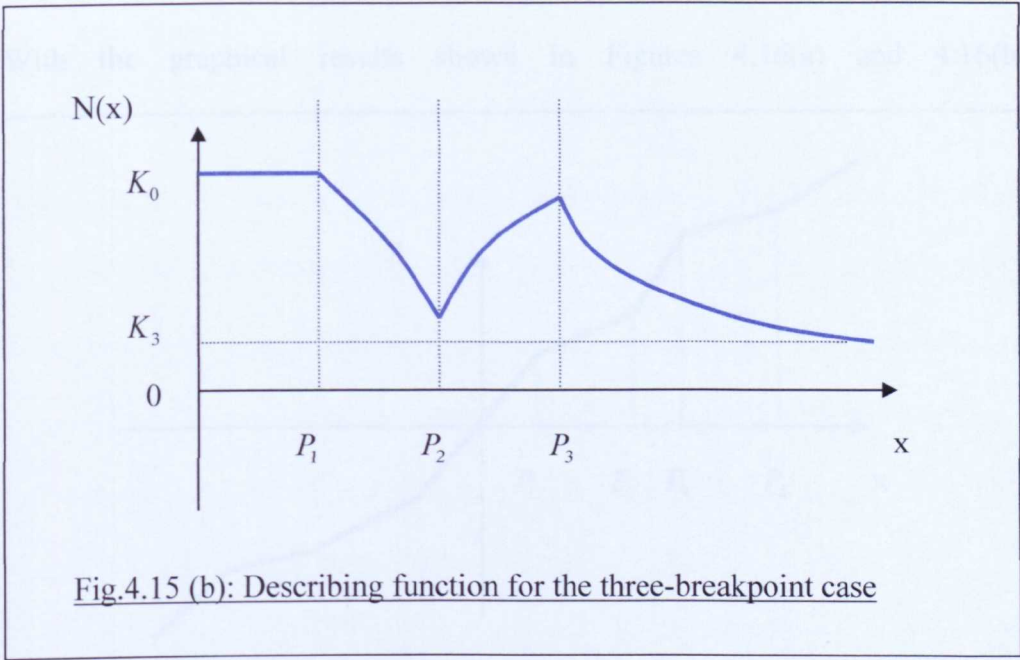
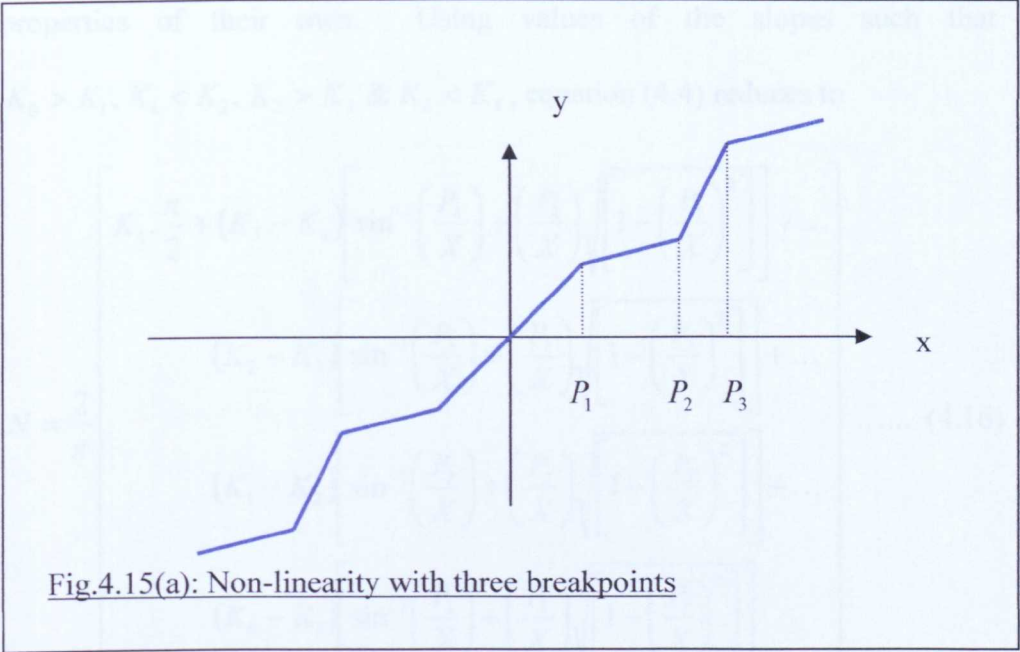
4.5.11 A non-linearity with three break-points (four slopes)

In this case only one condition is considered: $K_0 > K_1$ & $K_1 < K_2$ & $K_2 > K_3$.

Again, the results are obtained by using equation (4.4) to give

$$N = \frac{2}{\pi} \left[\begin{aligned} & K_3 \cdot \frac{\pi}{2} + (K_2 - K_3) \left[\sin^{-1} \left(\frac{P_3}{X} \right) + \left(\frac{P_3}{X} \right) \sqrt{1 - \left(\frac{P_3}{X} \right)^2} \right] + \dots \\ & (K_1 - K_2) \left[\sin^{-1} \left(\frac{P_2}{X} \right) + \left(\frac{P_2}{X} \right) \sqrt{1 - \left(\frac{P_2}{X} \right)^2} \right] + \dots \\ & (K_0 - K_1) \left[\sin^{-1} \left(\frac{P_1}{X} \right) + \left(\frac{P_1}{X} \right) \sqrt{1 - \left(\frac{P_1}{X} \right)^2} \right] \end{aligned} \right] \dots \dots (4.15)$$

The graphical results are shown in Figures 4.15(a) and 4.15(b). These results are used later in the investigation.



4.5.12 The four-breakpoint case

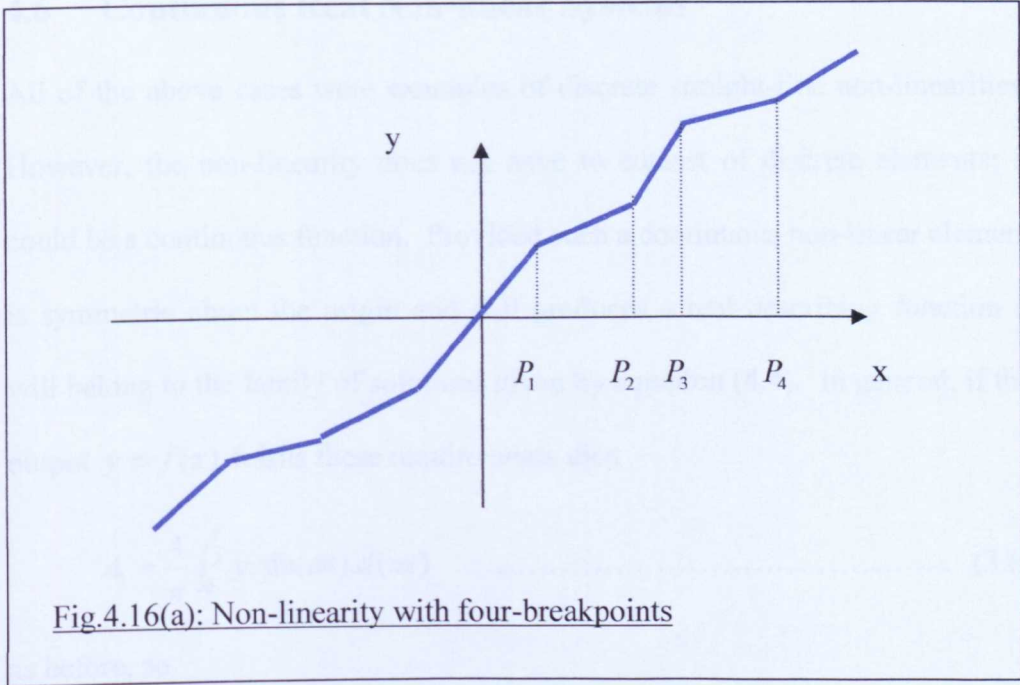
The final discrete case to be considered is the non-linearity with five straight sections and four breakpoints. Again, only one condition is being considered

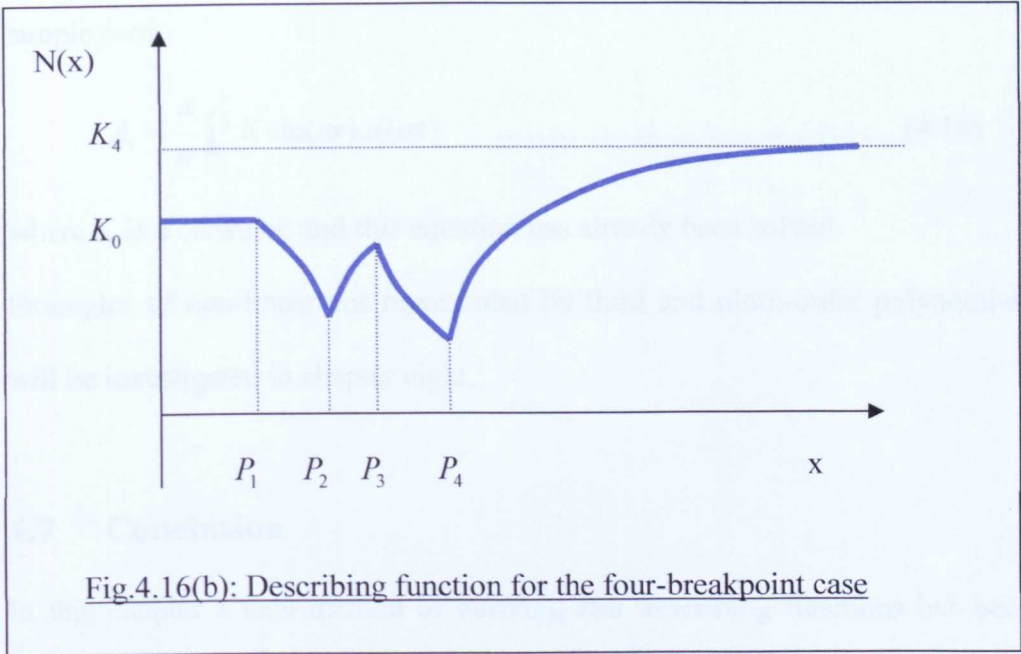
but later in the research that enables the crux of the investigation to be completed: Namely, the deliberate use of artificially created non-linearities to alter the behaviour of systems which contain inherent undesirable non-linear properties of their own. Using values of the slopes such that

$K_0 > K_1, K_1 < K_2, K_2 > K_3$ & $K_3 < K_4$, equation (4.4) reduces to

$$N = \frac{2}{\pi} \left[\begin{aligned} &K_4 \cdot \frac{\pi}{2} + (K_3 - K_4) \left[\sin^{-1} \left(\frac{P_4}{X} \right) + \left(\frac{P_4}{X} \right) \sqrt{1 - \left(\frac{P_4}{X} \right)^2} \right] + \dots \\ &(K_2 - K_3) \left[\sin^{-1} \left(\frac{P_3}{X} \right) + \left(\frac{P_3}{X} \right) \sqrt{1 - \left(\frac{P_3}{X} \right)^2} \right] + \dots \\ &(K_1 - K_2) \left[\sin^{-1} \left(\frac{P_2}{X} \right) + \left(\frac{P_2}{X} \right) \sqrt{1 - \left(\frac{P_2}{X} \right)^2} \right] + \dots \\ &(K_0 - K_1) \left[\sin^{-1} \left(\frac{P_1}{X} \right) + \left(\frac{P_1}{X} \right) \sqrt{1 - \left(\frac{P_1}{X} \right)^2} \right] \end{aligned} \right] \dots (4.16)$$

With the graphical results shown in Figures 4.16(a) and 4.16(b)





4.6 Continuous Real Non-linear Systems

All of the above cases were examples of discrete straight-line non-linearities. However, the non-linearity does not have to consist of discrete elements; it could be a continuous function. Provided such a continuous non-linear element is symmetric about the origin and still produces a real describing function it will belong to the family of solutions given by equation (4.4). In general, if the output $y = f(x)$ fulfils these requirements then

$$A_1 = \frac{4}{\pi} \int_0^{\frac{\pi}{2}} y. \sin(\omega t). d(\omega t) \dots\dots\dots (3.6)$$

as before, so

$$A_1 = \frac{4}{\pi} \int_0^{\frac{\pi}{2}} f(x) \sin(\omega t). d(\omega t) = \frac{4}{\pi} \int_0^{\frac{\pi}{2}} f(X. \sin(\omega t)) \sin(\omega t). d(\omega t) \dots\dots\dots (4.13)$$

The continuous function may, or may not, be integrable but if a large number, n , of piecewise elements of the non-linearity are considered then the individual elements $f(x_i)$ can be treated as constants and the integral reduces to the simple form

$$A_1 = \frac{4}{\pi} \int_0^{\frac{\pi}{2}} K \sin(\omega t) . d(\omega t) \dots\dots\dots (4.14)$$

where K is a constant, and this equation has already been solved.
Examples of non-linearities represented by third and ninth-order polynomials will be investigated in chapter eight.

4.7 Conclusion

In this chapter a new method of deriving real describing functions has been developed, together with an algorithm which enables the describing functions to be rapidly obtained. It has been shown to be applicable to continuous as well as to discontinuous states. As the method allowed real describing functions to be easily and rapidly obtained, the opportunity was taken to obtain a large range of describing functions for this particular family of non-linearities. The purpose of this was to enable various features to be seen which are not normally so obvious from just a few examples. Because of this approach it has become clear that many describing functions can be sketched intuitively by hand.

The next stage, which is dealt with in chapter five, has been to place various non-linearities in series with a transfer function which has been chosen as a standard for this investigation. These combinations have been placed, in turn,

in the forward path of a unit feedback circuit. The describing functions of standard non-linearities and then various custom-designed non-linearities have been calculated and the performance assessments used to develop prediction techniques for more sophisticated systems.

Chapter Five

**SIMULATION AND ANALYSIS OF NON-LINEAR
SYSTEMS**

5.1 Introduction

This chapter deals with the interaction of a group of non-linear systems, as represented by real describing functions, with the inverse Nyquist locus of a type 0, third-order, linear system with a transfer function is given in equation 5.1.

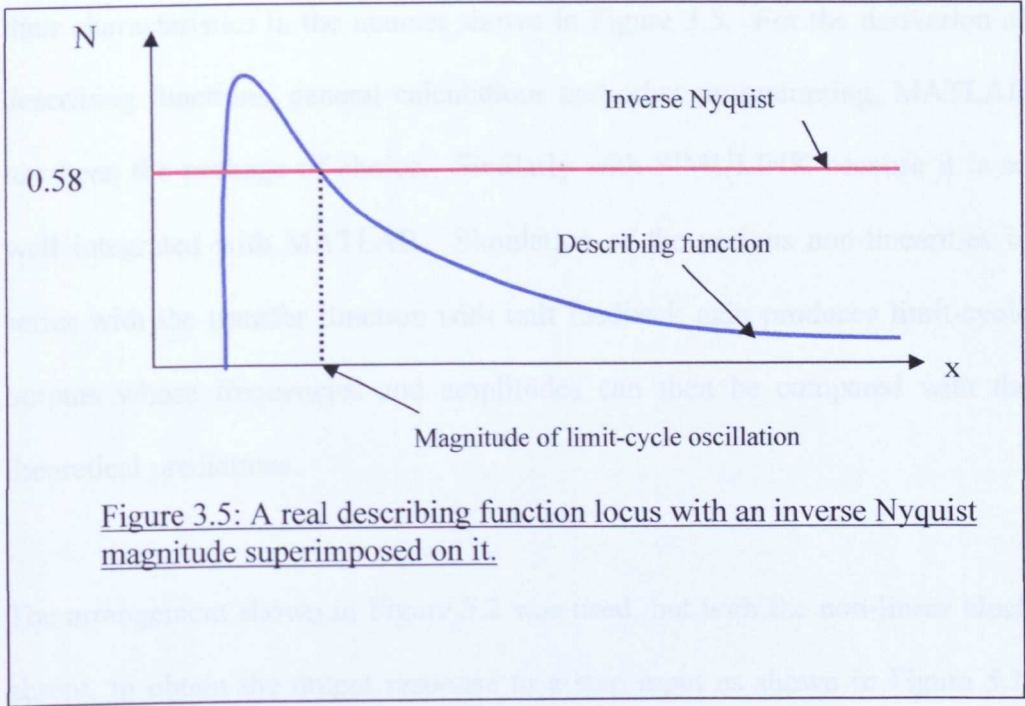
$$G(s) = \frac{K}{s^3 + 5s^2 + 6s + 1} \dots\dots\dots (5.1)$$

This is an adaptation of an example used by Kim, *et al.* (2000) which was the transfer function for an incubator used in a gynaecology ward. For the purposes of this investigation gain K had to be greater than 29 to ensure that the basic system would be unstable without any non-linear elements present. A value of K = 50 was chosen to give an unstable response and so ensure limit-cycle behaviour in appropriate cases without having to add extra gains to the non-linearities, and this gave a unit-feedback step response shown in Figure 5.1. The roots for this value of K are:

$$-5.5676, \quad +0.2838 + 3.0132j, \quad +0.2838 - 3.0132j$$

From the closed-loop characteristic equation the frequency of oscillation is 2.45 rad/s (to 3 significant figures) and is independent of K. If the magnitude of the inverse Nyquist is calculated for K = 50 with only the real terms present (i.e.: for the point where it crosses the real axis) then the cross-over point is

found to be -0.58 cm. This is the inverse Nyquist value shown in red in Figure 3.5 [reproduced here for clarity] and in all subsequent figures which plot the describing function/inverse Nyquist loci.



The relevant calculations are shown in Appendix A2.1.

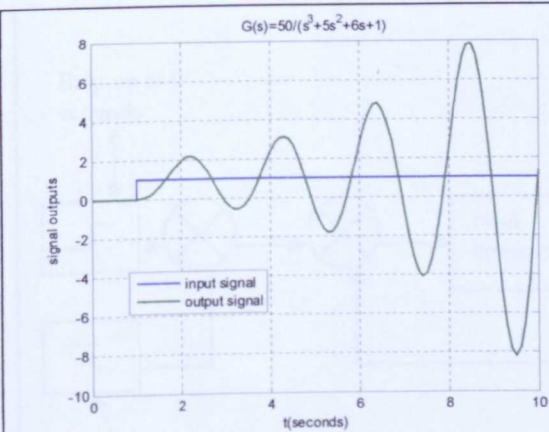
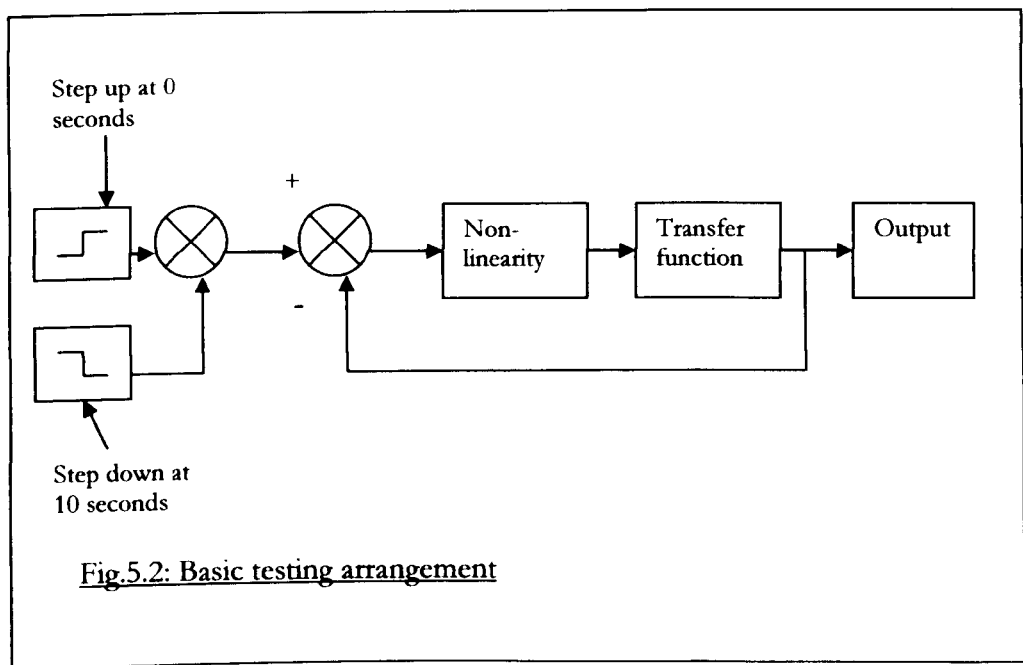


Fig. 5.1: Step response of standard transfer function with unit feedback

The method of analysis, based on Kochenburger's Stability Criterion (section 3.5), assumes that the non-linear element can be dealt with separately and its

describing function found. The describing functions for these non-linearities were then used in conjunction with the inverse Nyquist plot of the transfer function in equation 5.1 to predict when limit-cycles would occur and to obtain their characteristics in the manner shown in Figure 3.5. For the derivation of describing functions, general calculations and other programming, MATLAB has been the package of choice. Similarly with SIMULINK because it is so well integrated with MATLAB. Simulation of the various non-linearities in series with the transfer function with unit feedback gain produces limit-cycle outputs whose frequencies and amplitudes can then be compared with the theoretical predictions.

The arrangement shown in Figure 5.2 was used, but with the non-linear block absent, to obtain the output response to a step input as shown in Figure 5.1. The responses, with the non-linear block present, for all of the non-linear modules can then be compared with this result.



5.2 Testing the system with the standard non-linearities

As shown above, a value of K which made the basic system unstable so that no extra gains had to be added to the non-linearities, and then when these were present, if all other conditions are favourable, limit-cycles would occur.

A SIMULINK model was used, with various standard non-linearities from the toolbox inserted. The basic arrangement is shown in Figure 5.2.

5.2.1 Hard Saturation

When the saturation non-linearity was used, with a 10 second step as input, the oscillatory output shown in Figure 5.3 was produced. A step of two units was used so that it would comfortably pass into the saturation region. A vital check was run at this stage. This took the form of an open-loop simulation run with a ramp input and only the non-linearity in the forward path, Figure 5.4a. The output of this, Figure 5.4b gave the correct form (correct parameters) for the characteristic to be used when obtaining the describing function. The describing function for this non-linearity was calculated using a suitable *.m* file and the output then plotted with the magnitude of the Inverse Nyquist diagram superimposed on it to produce the result shown in Figure 5.5. From this, the expected magnitude of the limit cycle was then obtained and the result compared with the simulation in Figure 5.3.

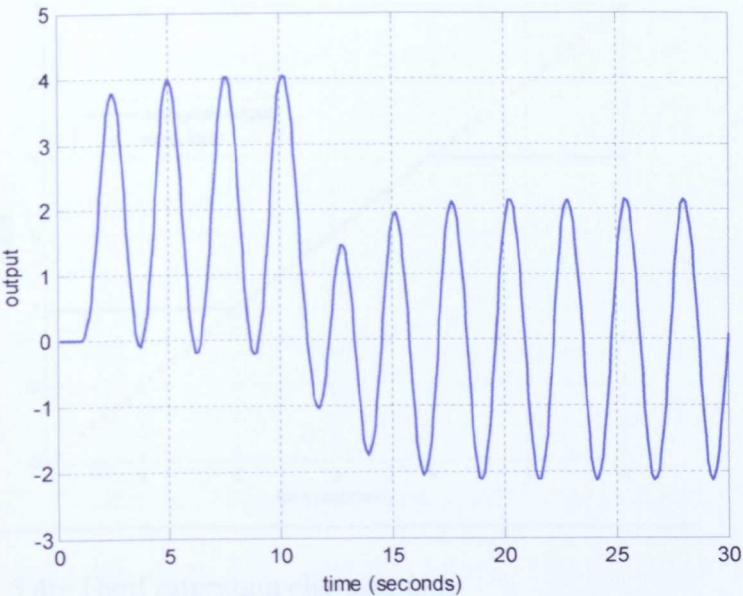


Fig. 5.3: Limit-cycle oscillation with hard saturation

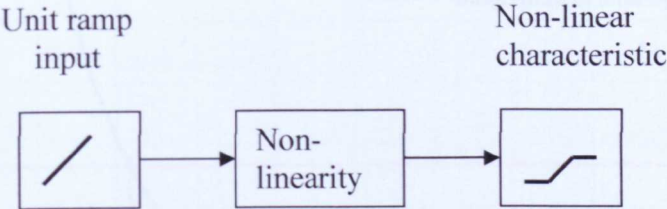


Fig. 5.4a: Circuit to display the non-linear characteristic

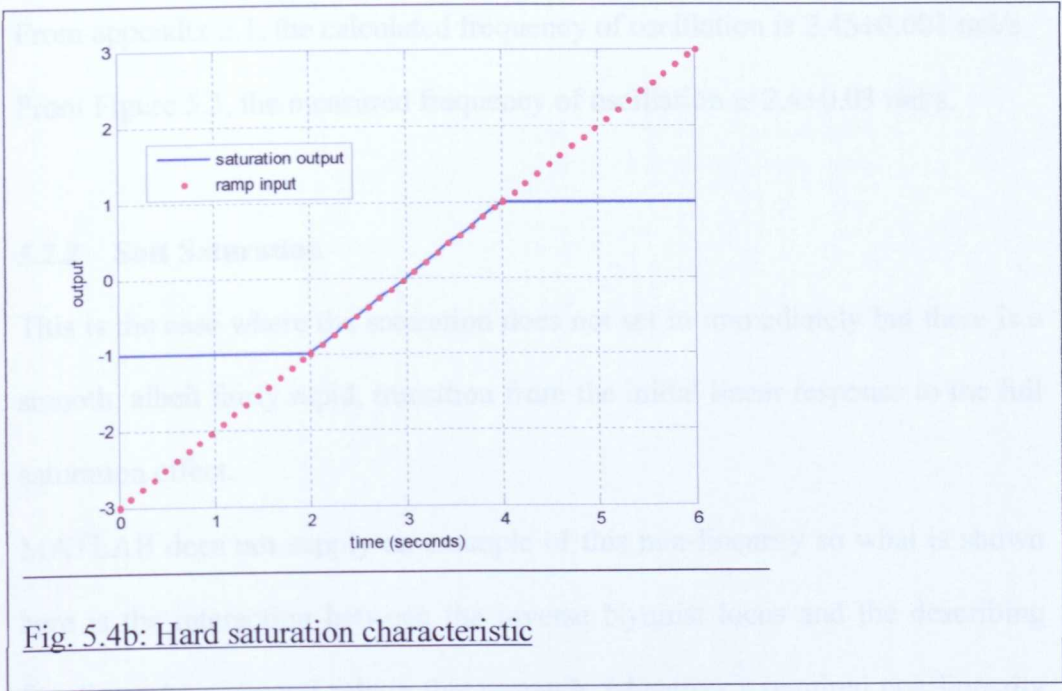


Fig. 5.4b: Hard saturation characteristic

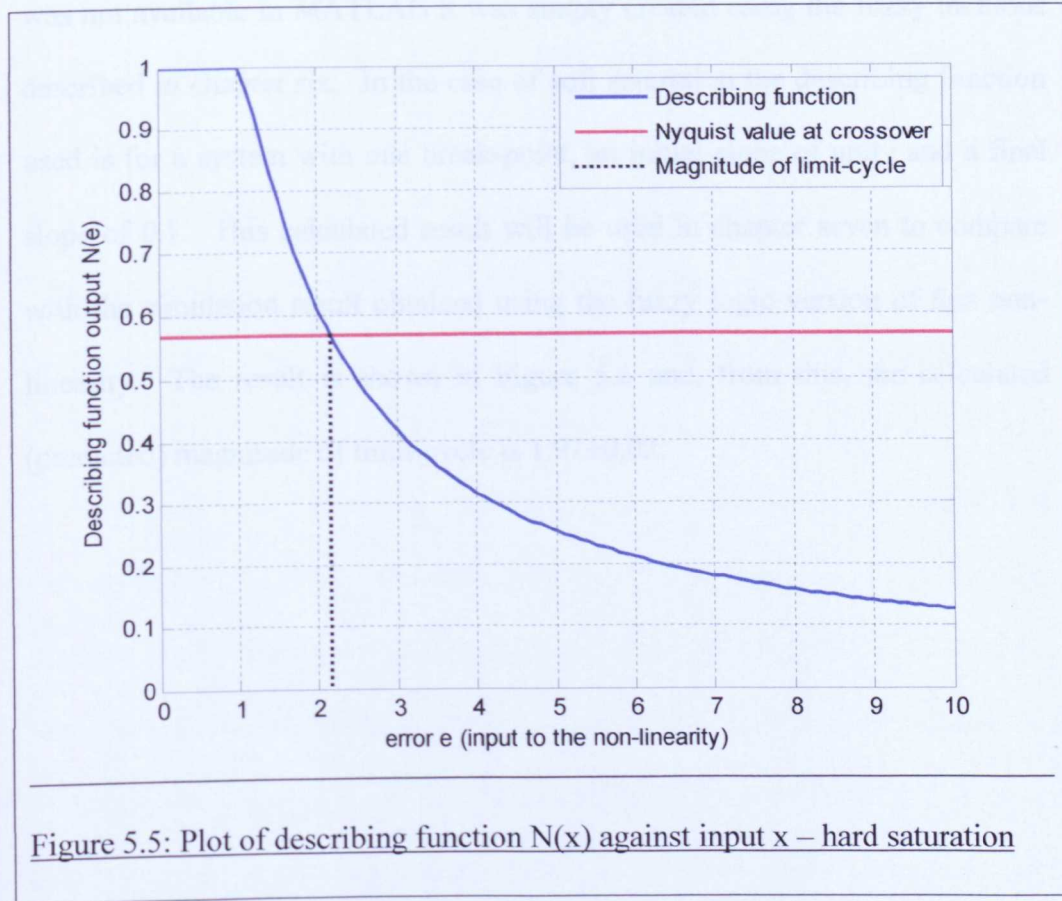


Figure 5.5: Plot of describing function $N(x)$ against input x – hard saturation

The calculated magnitude of the limit-cycle, from Figure 5.5, is 2.15 ± 0.02

The actual magnitude of the limit-cycle, from Figure 5.3, is 2.15 ± 0.03 .

From appendix 2.1, the calculated frequency of oscillation is 2.45 ± 0.001 rad/s.

From Figure 5.3, the measured frequency of oscillation is 2.4 ± 0.03 rad/s.

5.2.2 Soft Saturation

This is the case where the saturation does not set in immediately but there is a smooth, albeit fairly rapid, transition from the initial linear response to the full saturation effect.

MATLAB does not supply an example of this non-linearity so what is shown here is the interaction between the inverse Nyquist locus and the describing function. As a general rule in this research, whenever a required non-linearity was not available in MATLAB it was simply created using the fuzzy methods described in chapter six. In the case of soft saturation the describing function used is for a system with one break-point, an initial slope of unity and a final slope of 0.1. This calculated result will be used in chapter seven to compare with the simulation result obtained using the fuzzy logic version of this non-linearity. The result is shown in Figure 5.6 and, from this, the calculated (predicted) magnitude of limit-cycle is 1.91 ± 0.02 .

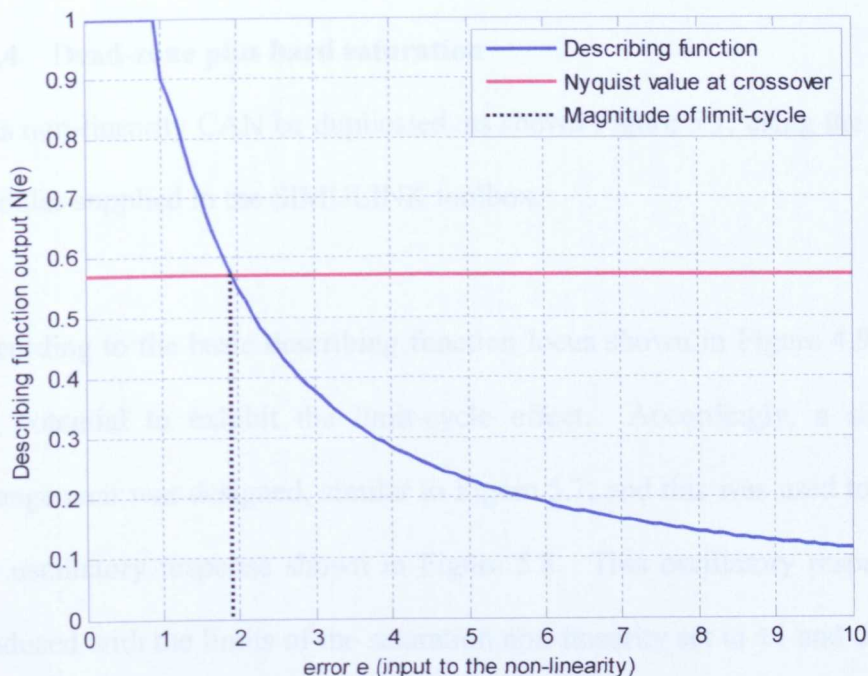


Figure 5.6: Plot of describing function $N(x)$ against input x – soft saturation

5.2.3 Dead-zone

This non-linearity is one of the standards available in SIMULINK and so its behaviour can be tested. However, on its own, it did not produce a stable limit-cycle effect. It simply went from over-damped decay of the signal at low gain settings, to runaway oscillation at higher gains. This was exactly what was expected when looking at the describing function in Figure 4.8. Provided the value of the inverse Nyquist at the crossing point was below the slope of the outer part of the non-linearity, so that the describing function locus crossed it, the locus would simply go from a region where the gain was less than unity, so that the input signal would simply decay, to a region where the overall gain was greater than unity. After that point, with no way back as far as the describing function was concerned, the oscillations would simply grow uncontrollably. The system would be unstable.

5.2.4 Dead-zone plus hard saturation

This non-linearity CAN be duplicated, as shown Figure 5.7, using the standard modules supplied in the SIMULINK toolbox.

According to the basic describing function locus shown in Figure 4.9 this has the potential to exhibit the limit-cycle effect. Accordingly, a simulation arrangement was designed, similar to Figure 5.7, and this was used to produce the oscillatory response shown in Figure 5.8. This oscillatory response was produced with the limits of the saturation non-linearity set to ± 1 and dead-zone set to ± 0.5 .

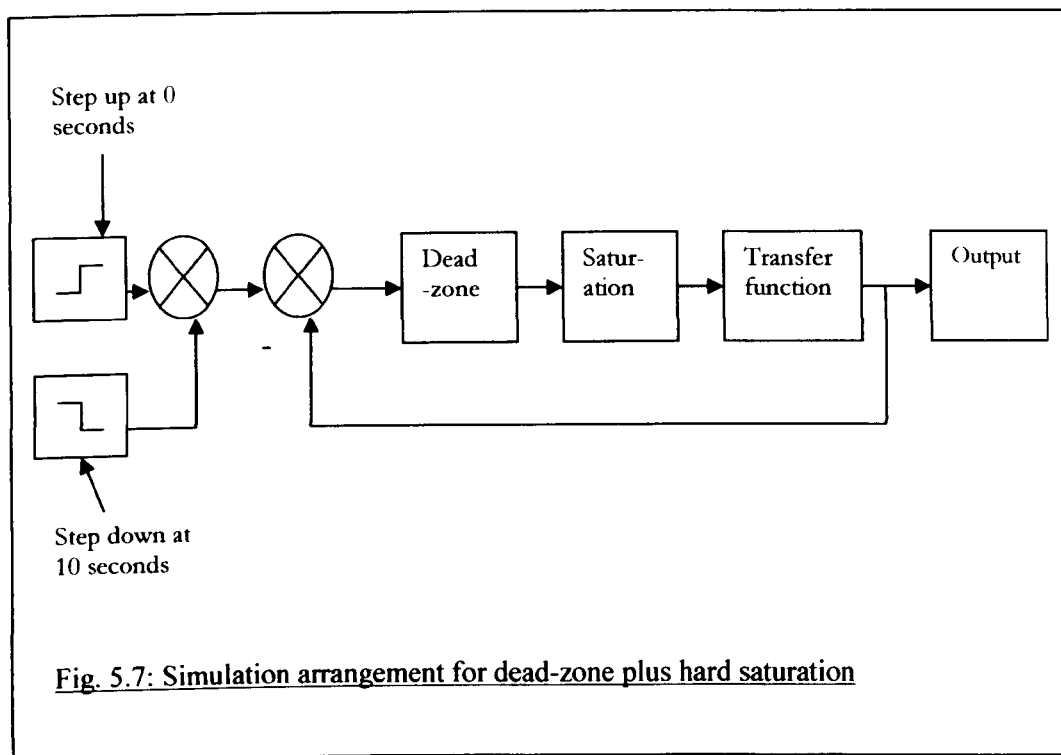


Fig. 5.7: Simulation arrangement for dead-zone plus hard saturation

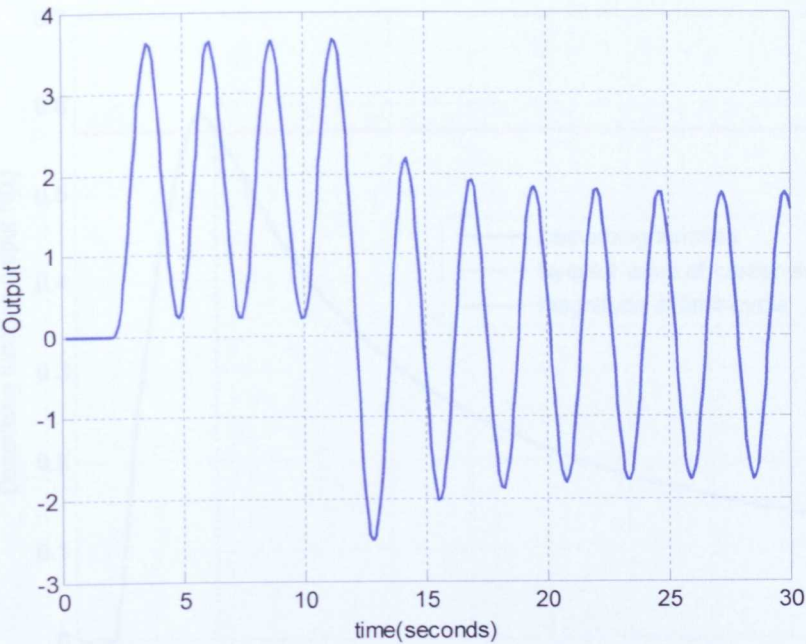


Fig. 5.8: Limit-cycle oscillation with hard saturation and dead-zone

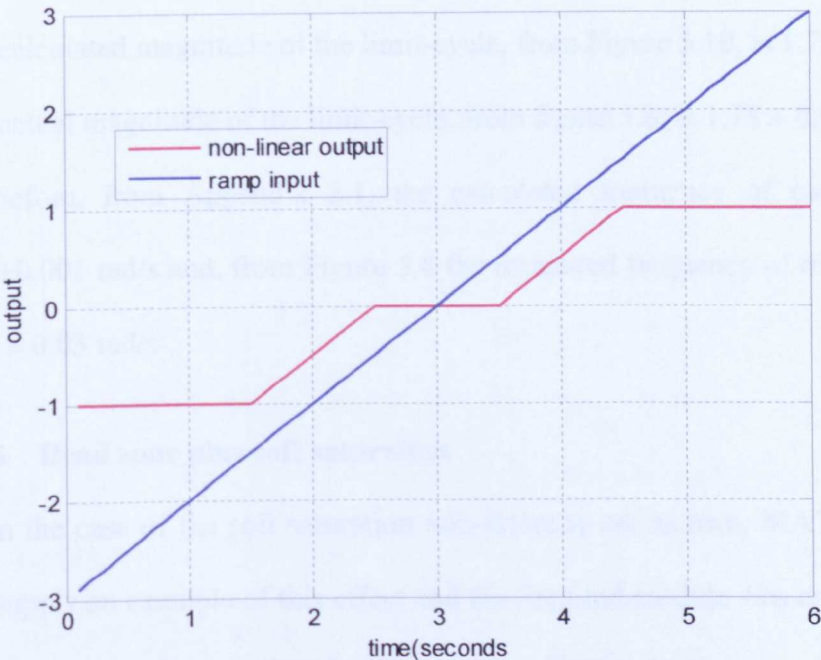


Fig. 5.9: Dead-zone + saturation characteristic

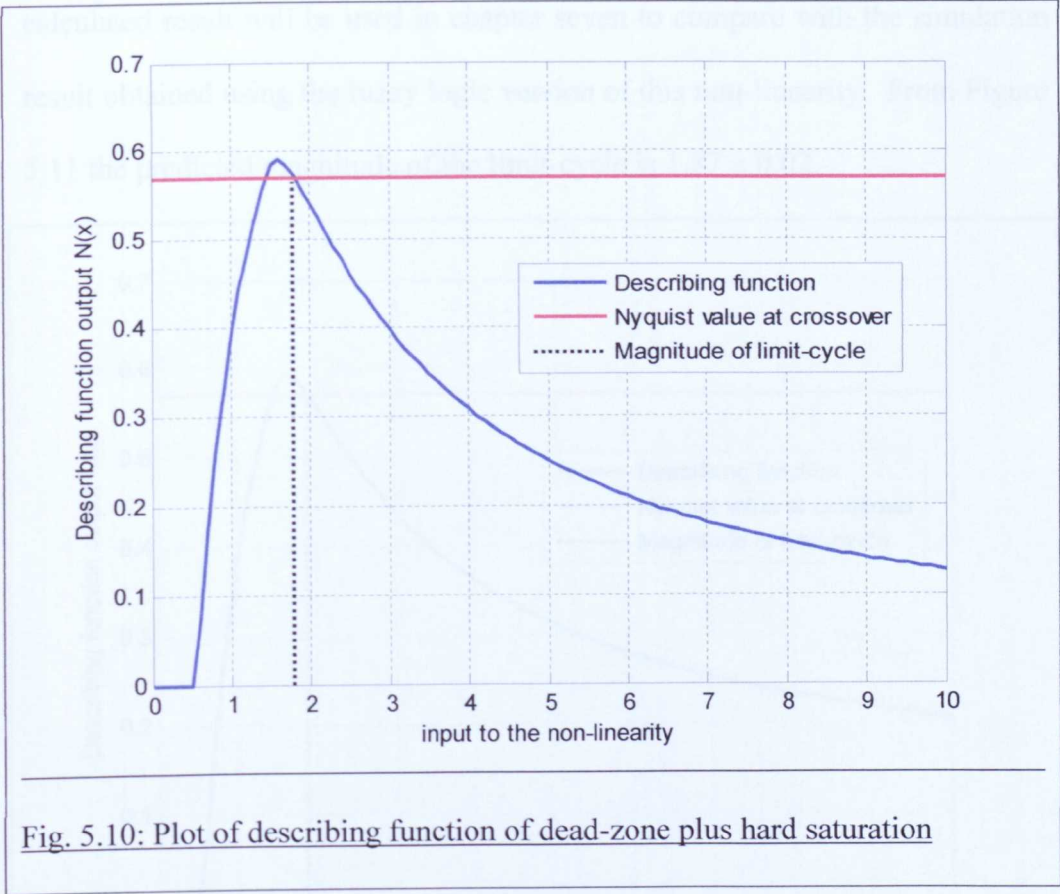


Fig. 5.10: Plot of describing function of dead-zone plus hard saturation

The calculated magnitude of the limit-cycle, from Figure 5.10, is 1.79 ± 0.02

The actual magnitude of the limit-cycle, from figure 5.8, is 1.78 ± 0.03

As before, from Appendix 2.1, the calculated frequency of oscillation is 2.45 ± 0.001 rad/s and, from Figure 5.8 the measured frequency of oscillation is 2.44 ± 0.03 rad/s

5.2.5 Dead zone plus soft saturation

As in the case of the soft saturation non-linearity on its own, MATLAB does not supply an example of this effect and the required module was created using the fuzzy methods described in chapter six. All that is shown at this stage is the interaction between the inverse Nyquist locus and the describing function. The describing function used is for systems with two break-points, an initial slope of zero, followed by a slope of one and a final slope of 0.1. Again, the

calculated result will be used in chapter seven to compare with the simulation result obtained using the fuzzy logic version of this non-linearity. From Figure 5.11 the predicted magnitude of the limit-cycle is 1.87 ± 0.02 .

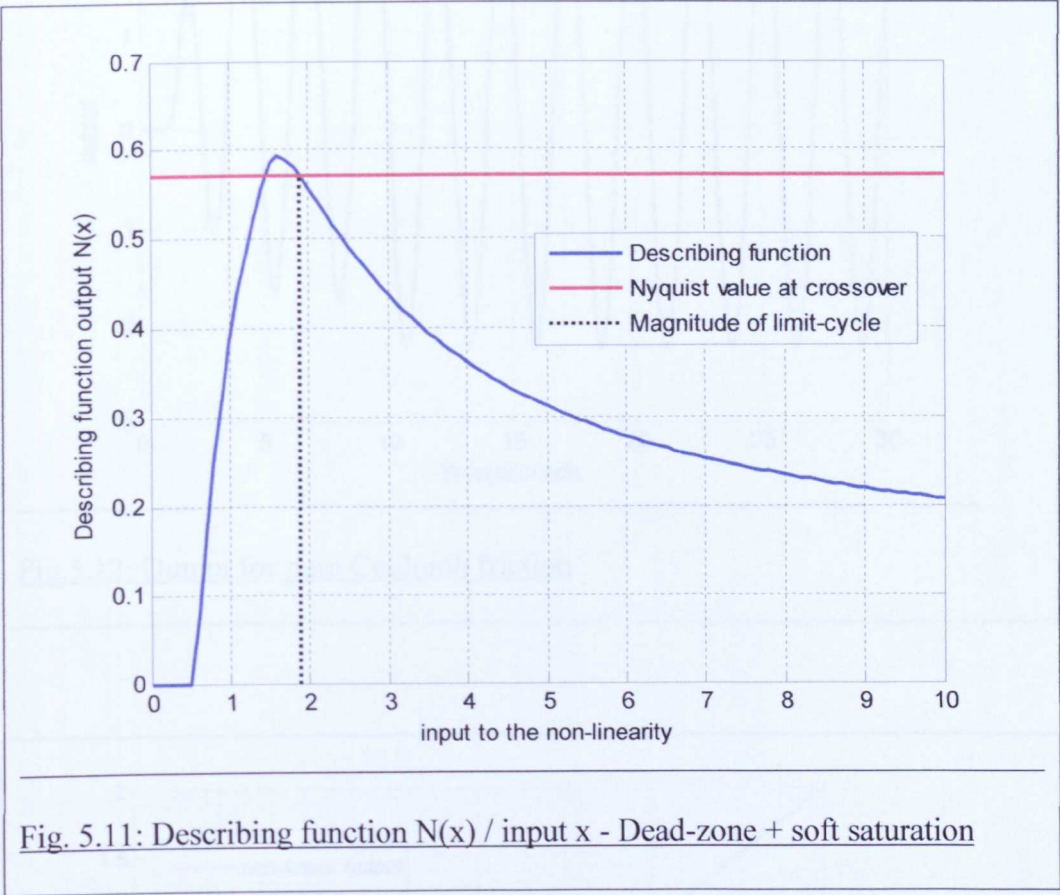


Fig. 5.11: Describing function $N(x) / \text{input } x$ - Dead-zone + soft saturation

5.2.6 The ideal relay (or pure Coulomb friction)

This gives a describing function of the form $y = \frac{1}{x}$ and consequently would cross the inverse Nyquist locus for any value of x , all decided by the value of the gain K of the linear part of the system. However, it is not possible to have a pure Coulomb effect on its own because the non-linearity must be some effect after time zero. The ideal relay is usually shown with saturation and that is what we have here.

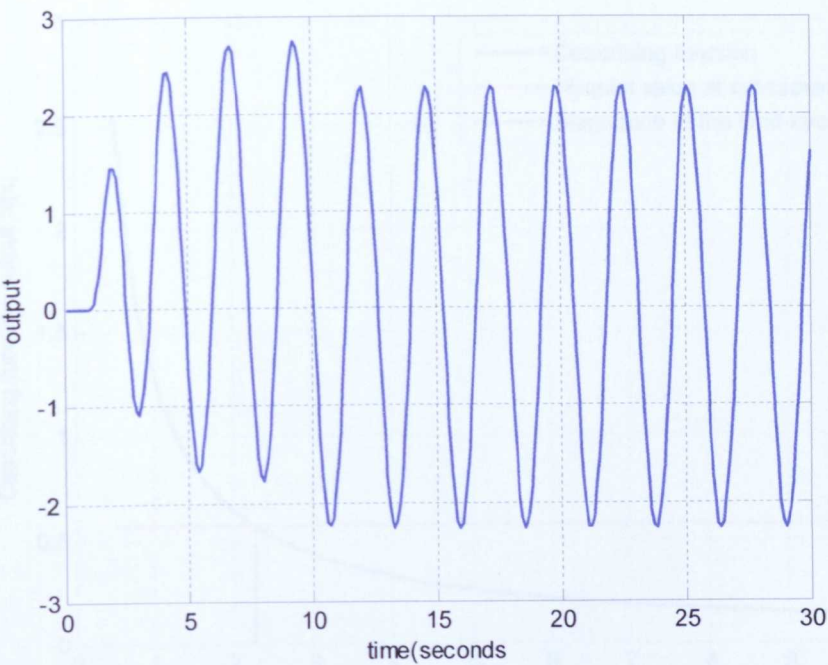


Fig.5.12: Output for pure Coulomb friction

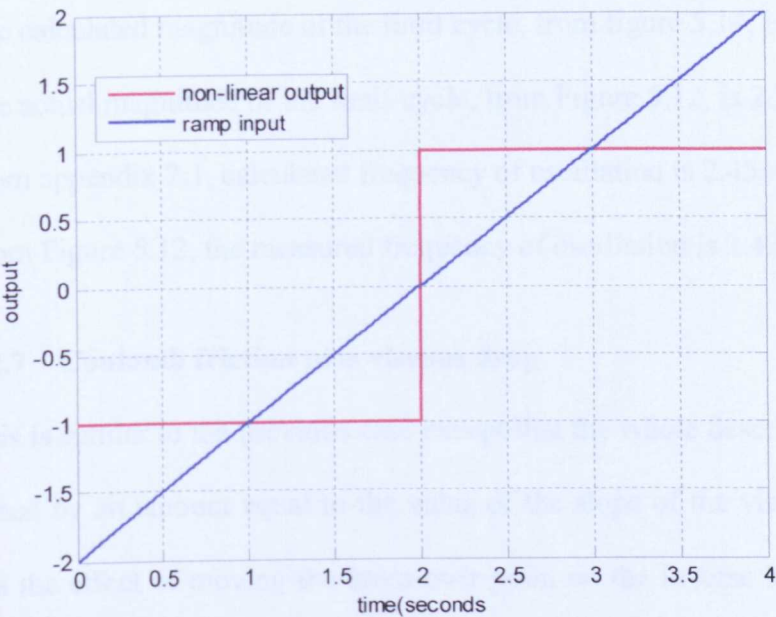


Fig. 5.13: Coulomb characteristic

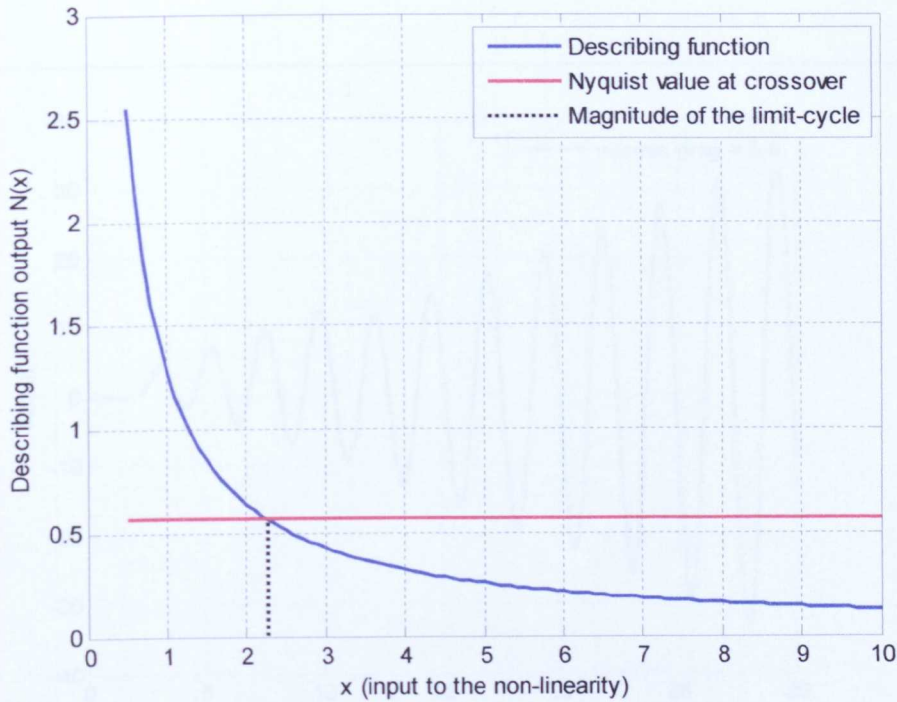


Fig. 5.14: Plot of describing function for Coulomb friction (ideal relay)

The calculated magnitude of the limit cycle, from figure 5.14, is 2.25 ± 0.02

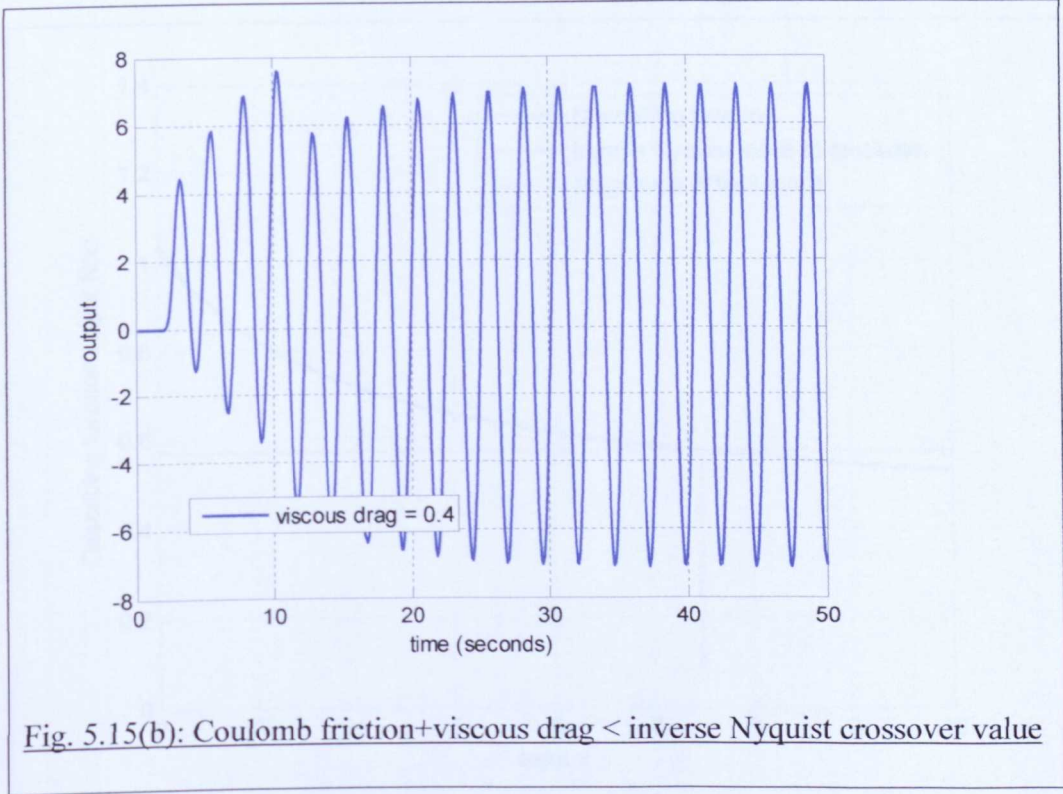
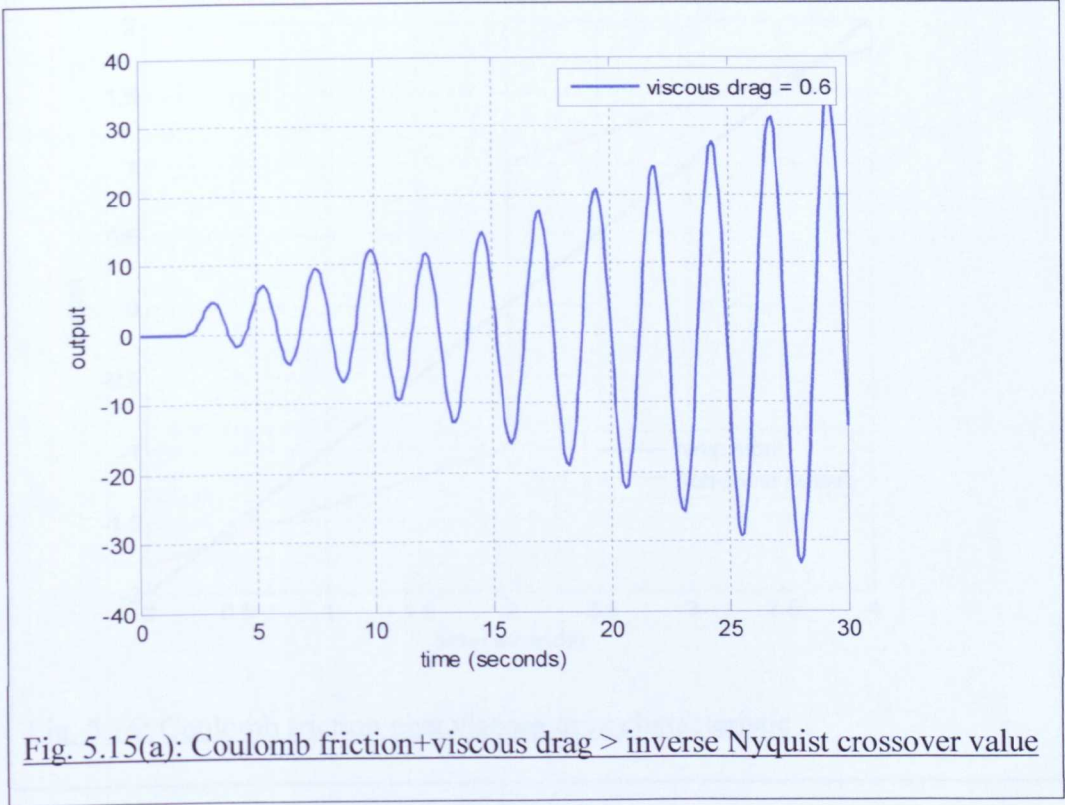
The actual magnitude of the limit-cycle, from Figure 5.12, is 2.30 ± 0.03

From appendix 2.1, calculated frequency of oscillation is 2.45 ± 0.001 rad/s

From Figure 5.12, the measured frequency of oscillation is 2.42 ± 0.03 rad/s

5.2.7 Coulomb friction plus viscous drag

This is similar to the previous case except that the whole describing function is raised by an amount equal to the value of the slope of the viscous drag. This has the effect of moving the cross-over point on the inverse Nyquist diagram, and hence the magnitude of the limit-cycle, to a higher value. Also, the gain of the viscous drag must be less than 0.58 (the value of the inverse Nyquist where it crosses the real axis) otherwise no limit-cycle will occur.



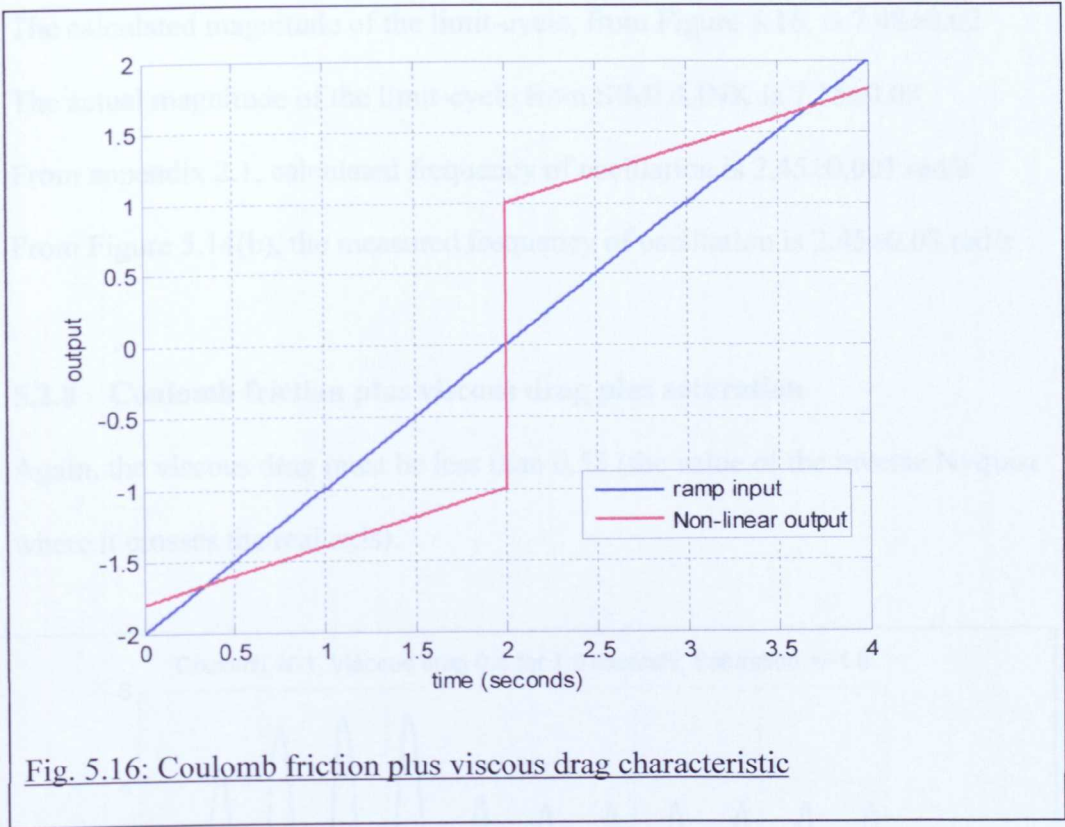


Fig. 5.16: Coulomb friction plus viscous drag characteristic

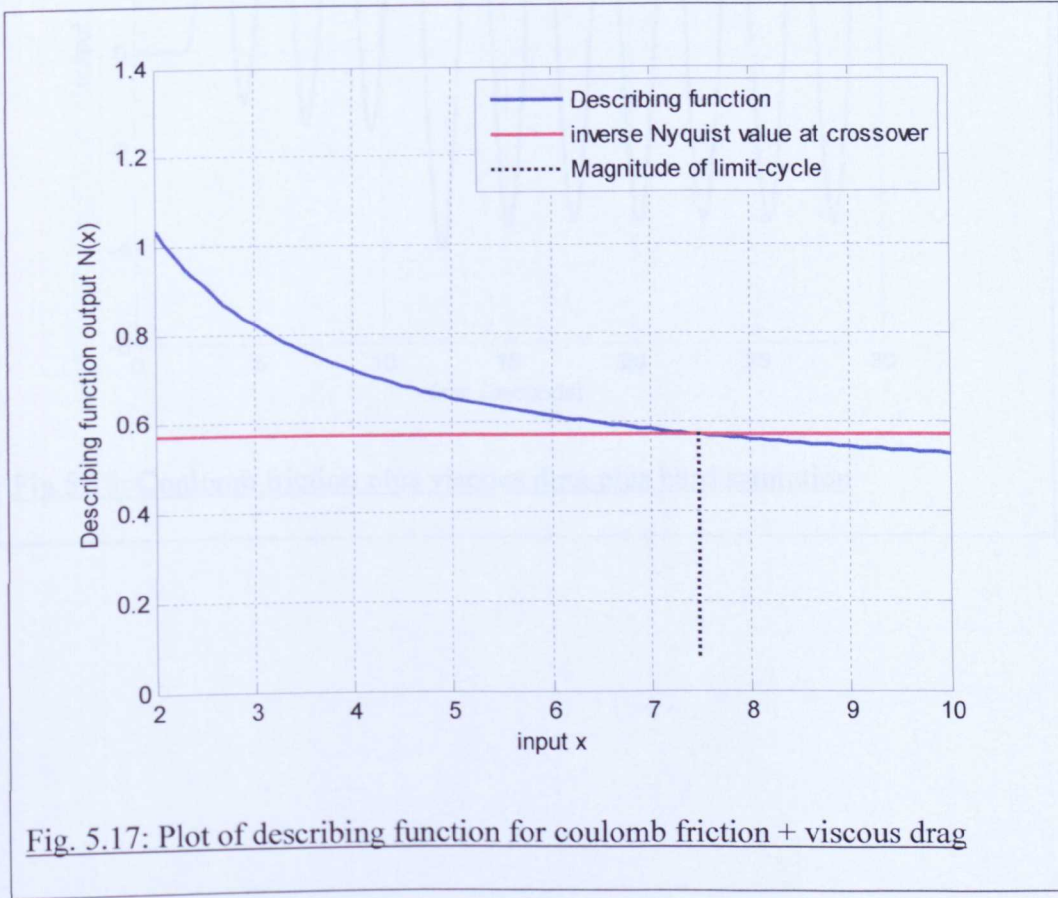


Fig. 5.17: Plot of describing function for coulomb friction + viscous drag

The calculated magnitude of the limit-cycle, from Figure 5.16, is 7.48 ± 0.02

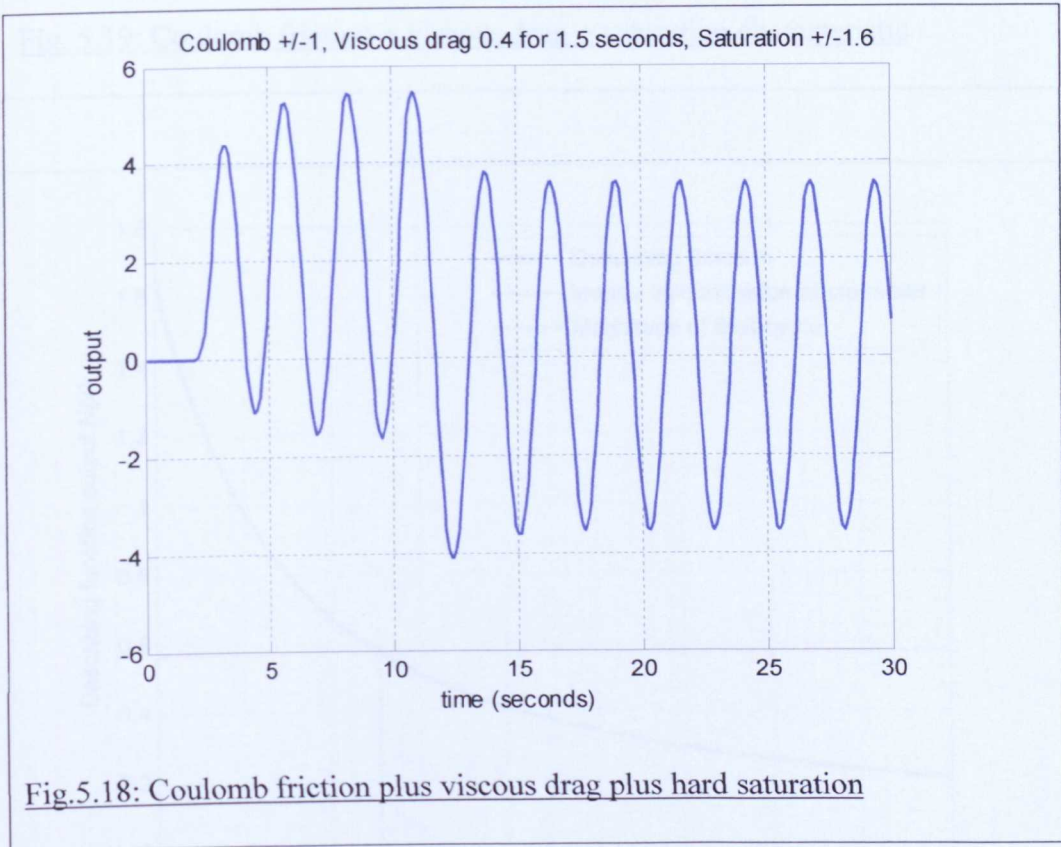
The actual magnitude of the limit-cycle from SIMULINK is 7.15 ± 0.03

From appendix 2.1, calculated frequency of oscillation is 2.45 ± 0.001 rad/s

From Figure 5.14(b), the measured frequency of oscillation is 2.45 ± 0.03 rad/s

5.2.8 Coulomb friction plus viscous drag plus saturation

Again, the viscous drag must be less than 0.58 (the value of the inverse Nyquist where it crosses the real axis).



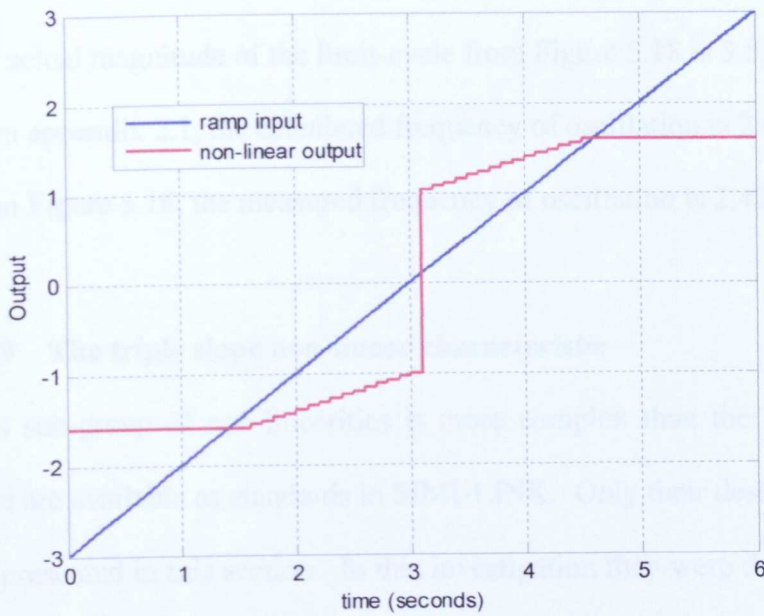


Fig. 5.19: Coulomb friction + viscous drag + saturation characteristic

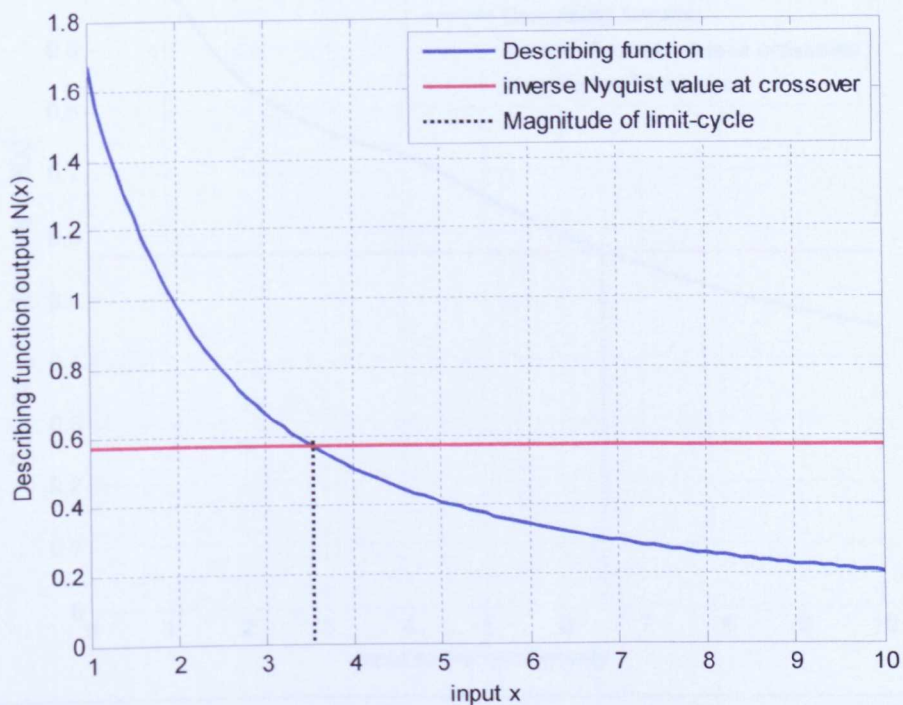


Fig. 5.20: Describing function for Coulomb friction + viscous drag + saturation

The calculated magnitude of the limit-cycle, from Figure 5.20, is 3.53 ± 0.02

The actual magnitude of the limit-cycle from Figure 5.18 is 3.52 ± 0.03

From appendix 2.1, the calculated frequency of oscillation is 2.45 ± 0.001 rad/s

From Figure 5.18, the measured frequency of oscillation is 2.42 ± 0.03 rad/s.

5.2.9 The triple slope non-linear characteristic

This sub-group of non-linearities is more complex than the earlier ones and none are available as standards in SIMULINK. Only their describing functions are presented in this section. In this investigation they were designed by fuzzy logic methods and their performances are demonstrated in chapter seven.

Case (i): $K_0 > K_1 > K_2$

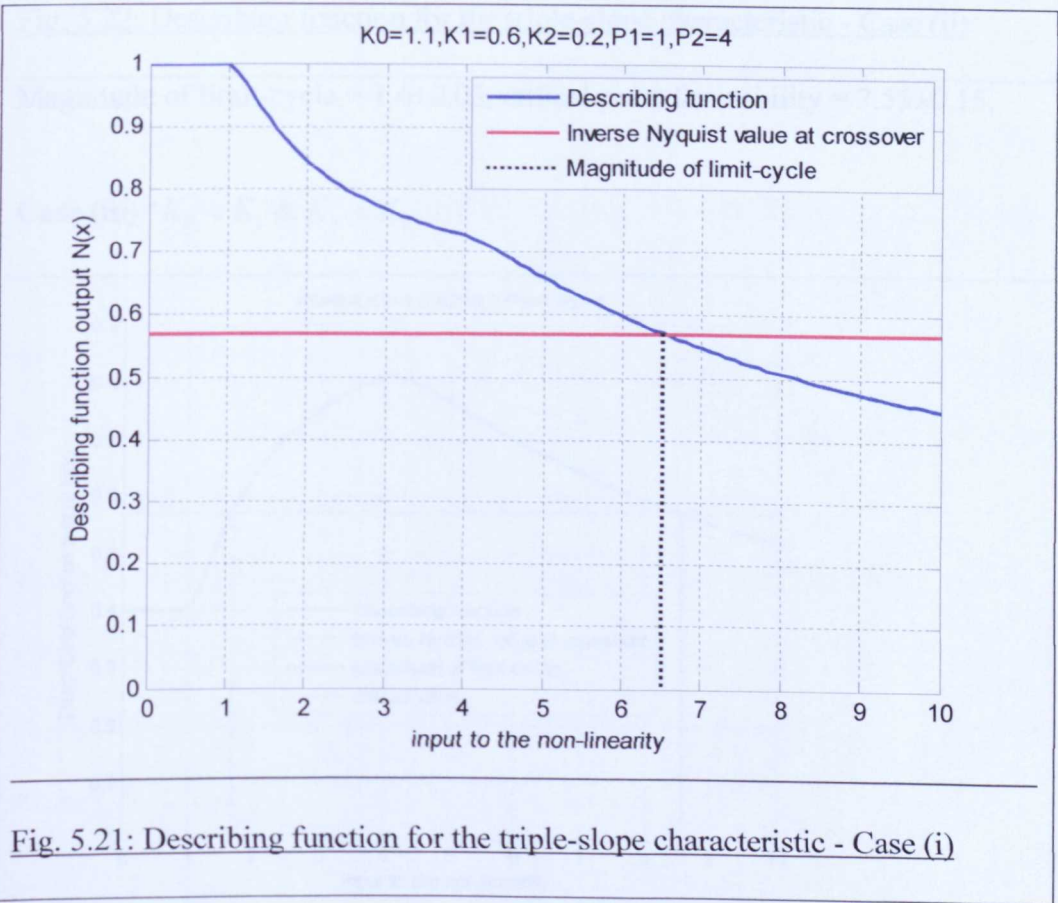
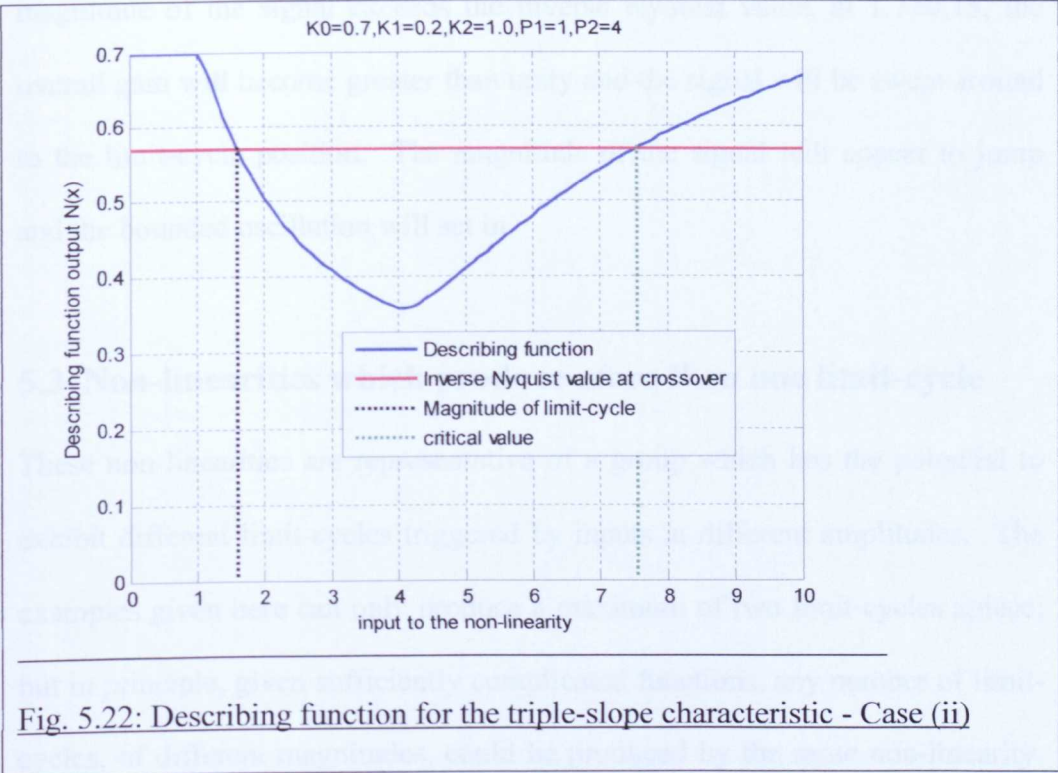


Fig. 5.21: Describing function for the triple-slope characteristic - Case (i)

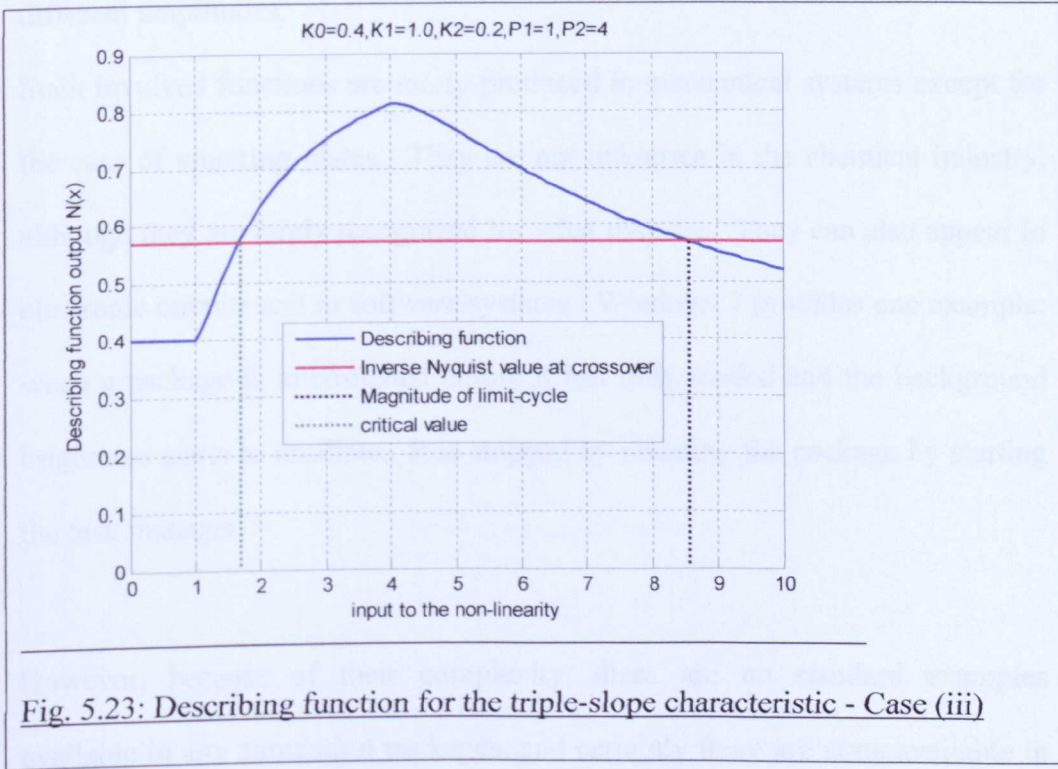
Magnitude of limit-cycle = 6.5 ± 0.05

Case (ii): $K_0 > K_1$ & $K_2 > K_1$



Magnitude of limit-cycle = 1.6 ± 0.05 , critical point for stability = 7.55 ± 0.15 .

Case (iii) $K_0 < K_1$ & $K_2 < K_1$



The magnitude of the limit-cycle = 8.55 ± 0.15 . It is worth noting that when the magnitude of the signal exceeds the inverse Nyquist value, at 1.7 ± 0.15 , the overall gain will become greater than unity and the signal will be swept around to the limit-cycle position. The magnitude of the signal will appear to jump and the bounded oscillation will set in.

5.3 Non-linearities which produce more than one limit-cycle

These non-linearities are representative of a group which has the potential to exhibit different limit-cycles triggered by inputs at different amplitudes. The examples given here can only produce a maximum of two limit-cycles apiece; but in principle, given sufficiently complicated functions, any number of limit-cycles, of different magnitudes, could be produced by the same non-linearity. This would carry across to the complex non-linearities which would have the potential to produce multiple limit-cycles with different frequencies as well as different amplitudes.

Such involved functions are rarely produced in mechanical systems except for the case of vibrating plates. They are not unknown in the chemical industry, although they are rarely recognized for what they are. They can also appear in electronic circuits and in software systems. Windows 7 provides one example: when a package is interrogated before it has fully loaded and the background brightness starts to oscillate. It is stopped by resetting the package by starting the task manager.

However, because of their complexity, there are no standard examples available in any simulation packages, and certainly there are none available in

SIMULINK. For this reason, only the describing functions, and their associated predictions, are presented in this section. The actual functioning non-linearities have been designed by fuzzy logic methods, and their behaviour demonstrated, in chapter seven. This is a further advantage of the design-technique that has been developed for use in this investigation. Apart from a comment in Atherton (1981) to the effect that since a particular differential equation may have more than one limit-cycle solution they may exist one inside another in a phase-plane portrait there are few other mentions in the literature, (Bendixson, 1901). To the best of the author's knowledge, no detailed investigations of their interactions have been reported in the research literature.

5.3.1 A non-linearity with three break-points (four slopes)

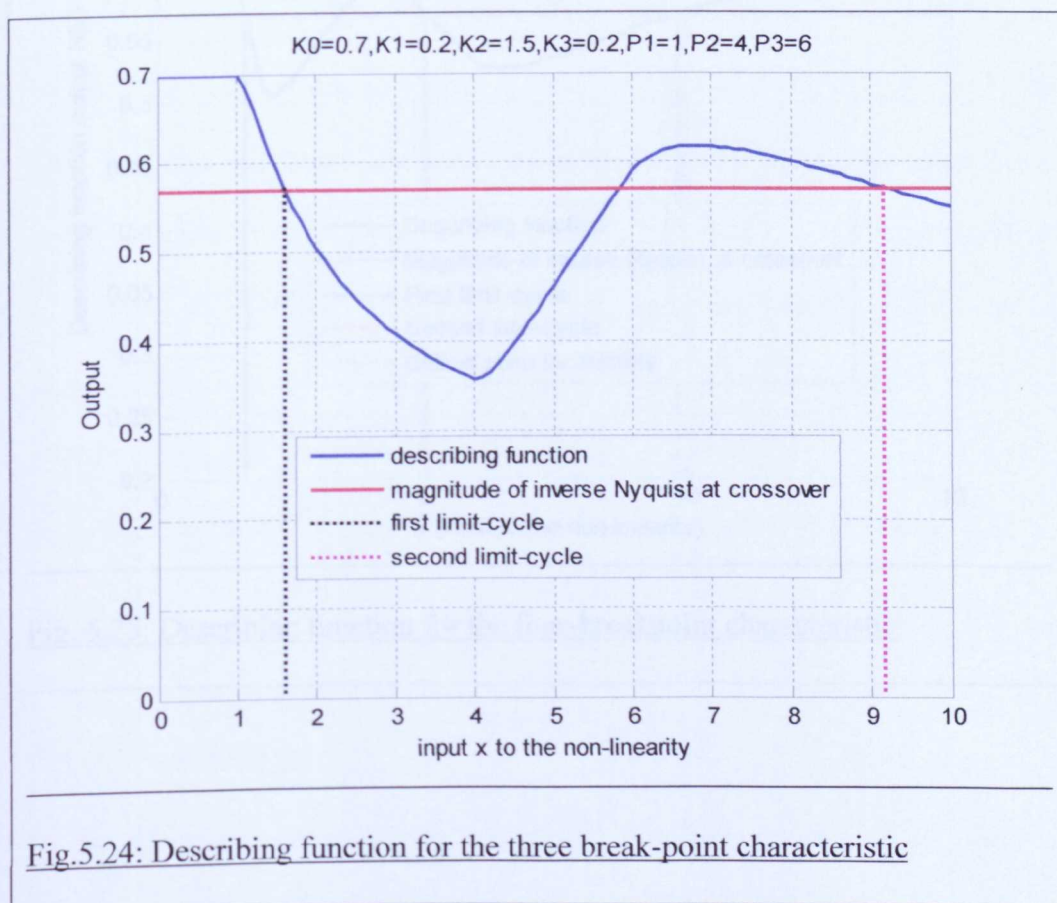


Fig.5.24: Describing function for the three break-point characteristic

This time there are two positions where limit-cycles might occur: (i) first limit cycle at 1.60 ± 0.05 , second limit-cycle at 9.15 ± 0.08 .

5.3.2 The four break-point case (five slopes)

This characteristic presents a slightly different situation to the previous case. Again, there are two positions at which a limit-cycle may occur, the first limit-cycle at 1.64 ± 0.05 and the second at 5.08 ± 0.05 but looking at the extreme right of the graph a critical point is marked.

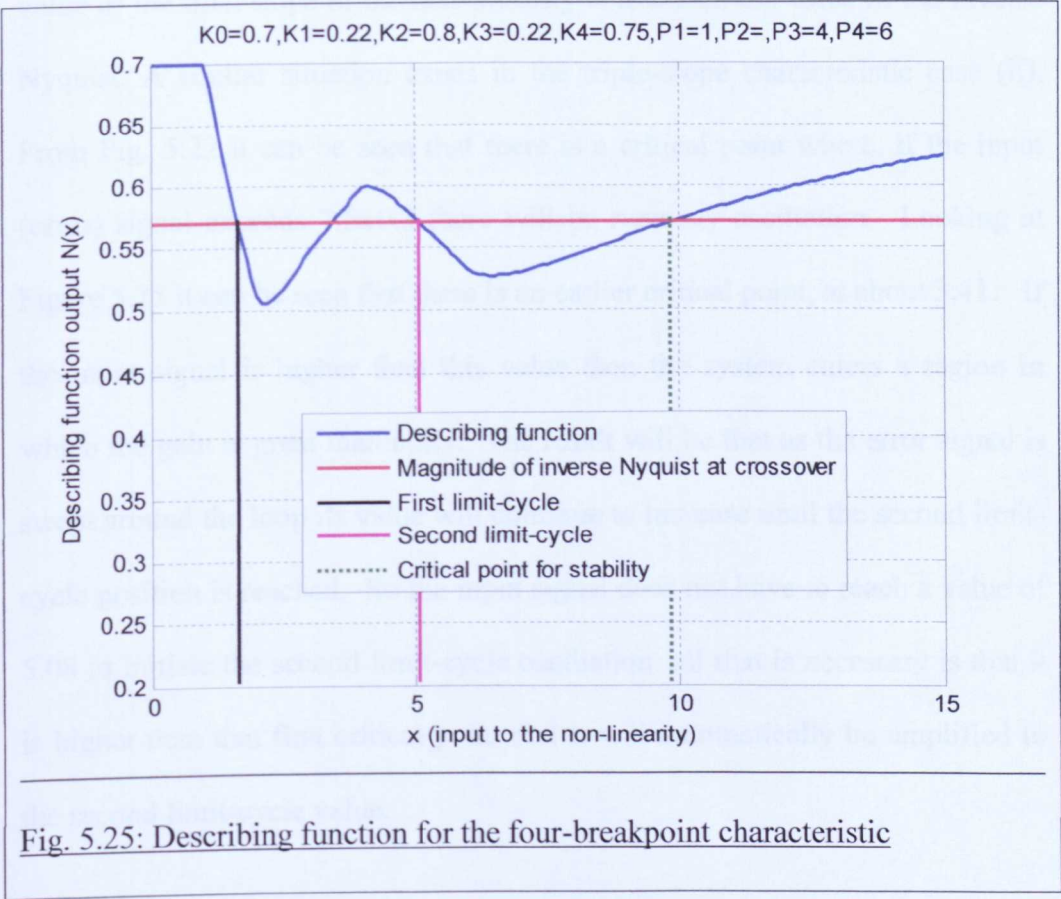


Fig. 5.25: Describing function for the four-breakpoint characteristic

5.4 The significance of critical points

In Figure 5.25, it can be seen that there is a critical value for the error signal, 9.85 ± 0.3 in this case, in which a rising value of the describing function may cross the inverse Nyquist locus a second time. This holds out the potential for instability if the input signal rises high enough (higher than this value). If using the describing function as a design tool, it can be seen from chapter four that the ends of the describing loci are asymptotic to the value of final slope. So to prevent runaway oscillation all that is needed is to make sure that the value of the final slope of the non-linearity is less than the value of the inverse Nyquist. A similar situation exists in the triple-slope characteristic case (ii). From Fig. 5.22 it can be seen that there is a critical point where, if the input (error) signal exceeds 7.5 ± 0.3 there will be runaway oscillation. Looking at Figure 5.25 it can be seen that there is an earlier critical point, at about 3.41. If the input signal is higher than this value then the system enters a region in which the gain is great than unity. The result will be that as the error signal is swept around the loop its value will continue to increase until the second limit-cycle position is reached. So the input signal does not have to reach a value of 5.08 to initiate the second limit-cycle oscillation; all that is necessary is that it is higher than that first critical point and it will automatically be amplified to the second limit-cycle value.

5.5 Conclusions

In this chapter the transfer function which had been chosen to be used, in conjunction with which all of the non-linearity performances were to be

judged, was introduced. The describing functions for the standard non-linearities were then obtained and any limit-cycles and their characteristics were predicted. These were then compared with simulated results.

More involved non-linear systems were then designed and their performances predicted. These designs culminated in systems which were predicted to produce multiple limit-cycle. However, since none of these non-linearities were part of the standard range supplied by SIMULINK, the checking of their performances against actual simulations has been postponed until chapter seven. Then they can be designed using fuzzy-logic techniques. Before this, the methods of fuzzy logic have to be introduced in chapter six.

The final item discussed was the concept of critical points, as defined in this chapter. Their relevance will be examined in chapters seven and eight.

Chapter Six

FUZZY SYSTEMS

6.1 Introduction

Fuzzy logic theory was developed in the 1960s (Zadeh, 1965, 1969) but it was not until the 1970s that the concepts were applied to control engineering (Mamdani, 1974; Mamdani *et al.*, 1975). Since then, fuzzy logic theory in all of its manifestations has continued to gain momentum. Kandel (1982) listed over 3000 “Key references in fuzzy pattern recognition”. In the journal for Fuzzy Sets and Systems over 5000 papers have been published so far; and similar numbers of papers have been published in other journals such as the IEEE Transactions in Fuzzy Logic. Zimmerman (1996) gave a database figure of 12000 published papers. Although fuzzy set theory was never a universal panacea that could solve all problems it has proved that it has considerable potential for a host of practical applications

For the current research the attraction of fuzzy control systems is that they are inherently non-linear and can themselves exhibit the range of features of classical non-linear systems. Furthermore, fuzzy logic techniques provide a very flexible method of modifying the actual shapes of signals by design. There are many other methods which are available to do this but when it comes to subtly designing partial inverses of a system, or to designing systems with several non-linearities then the fuzzy approach is superior.

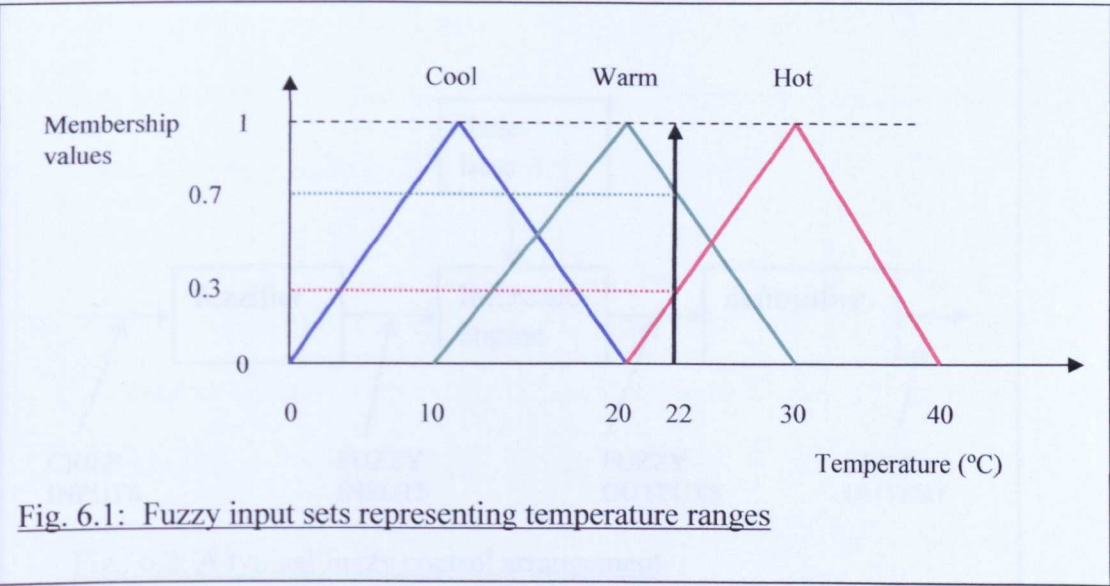
In this chapter the basic principles of fuzzy logic and of fuzzy control systems are briefly outlined. The possible alternative methods of altering the output signal are explained and reasons given for the approach chosen in this research.

6.2 Fuzzy Control Systems

The standard fuzzy controller design stems from the original one developed by Mamdani. He used a signal and its derivative as the inputs but this can be generalized to a group of input signals, each signal representing a different physical quantity. As an example, a two-input signal system, with a single output, is considered. Firstly, the crisp inputs have to be converted to fuzzy signals. This is accomplished by defining a group of fuzzy sets for each input variable, each set representing a range of the same physical variable as the input crisp signal but also representing the probability that the signal is represented in that range. For example, temperature might be represented by three sets, each representing a range of temperatures and each set labelled by a subjective term: *cold*, *warm* and *hot* as in Figure 6.1. The shape of the fuzzy set can be quite arbitrary, within certain bounds, but a triangular shape is the most easy to use because the membership values are most easily calculated for that shape. A related point is that triangular sets do not introduce any distortions or biases into the fuzzy calculations. In our example, a given crisp temperature value, 22°C say, would appear to intersect the fuzzy set labelled *warm* at the 0.7 mark, whilst it would cut the set labelled *hot* at 0.3. The significance of the membership function figures is that they represent the degree of confidence that an element which happened to be within that set is entirely a full-blown member of it i.e.: 100% membership, probability of

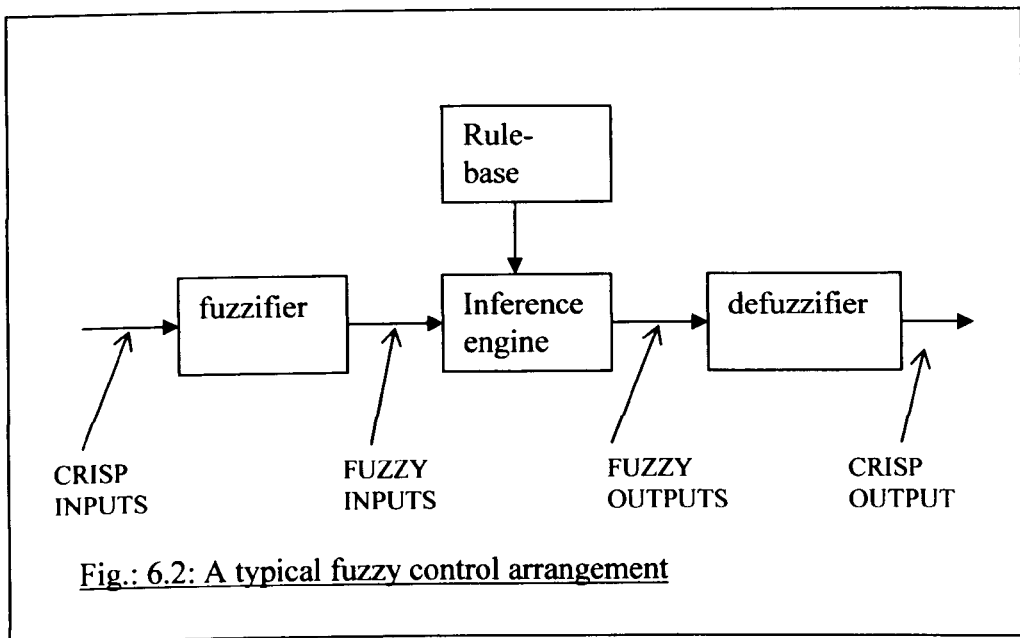
membership =1 (which is the maximum it can be), or perhaps is partly a member of some other set. In this example the single crisp temperature value 22°C is represented by two singleton fuzzy quantities: set Hot (22/0.3) and set Warm (22/0.7). The first number in brackets represents the element itself (its value or name) and the second number is its membership value for that set.

In general, this process of fuzzification of real, crisp, quantities produces two fuzzy values for each crisp value if there is overlap of the defined fuzzy sets, or one fuzzy value if there is no overlap but the defined fuzzy sets are just touching.



The next stage is to create a rule-base which looks at each fuzzy value for a given physical quantity and compares it with each respective fuzzy value for the incoming signal value for every other physical quantity. The outcome of the series of comparisons is formulated as a rule base which states what the output signal from the fuzzy controller, for that group of inputs, should be. This rule base is utilised by the inference engine which also applies appropriate

weighting to the individual output signals depending upon the membership values of the inputs. When all the fuzzy input signals at a given time-sample have been cross-correlated in this fashion their collective output results are then combined in some suitable fashion, in the defuzzification section, to give a single crisp output signal. What methods are employed to calculate the crisp output signal depends on the type of rule-base being used and sometimes on other factors such as the amount of computing power available or the purpose of the control system in use. The overall format of the basic fuzzy controller is shown in Figure 6.2.



6.3 Methods for tuning the fuzzy control systems

Currently there are four known ways in which the output signal of a fuzzy-logic controller can be adjusted, or tuned: (i) by adjustment of the fuzzy sets themselves (Ross, 2004) (ii) by altering the rule-base (Kickert *et al.*, 1978) (iii) by using an inference filter (Kiendl, 1994, 1998; Liming Hu *et al.*, 2007; Güler

et al., 2005; Mendoza *et al.*, 2007) or (iv) by using negative as well as positive rules (Kiendl, 1998; Krone *et al.*, 1996; Branson *et al.*, 2001).

- (i) Various shapes can be used to define the fuzzy sets and can have a significant effect in classification and cluster analysis but for control purposes they have little effect on overall performance (Mansoor *et al.* 2007; Mitiam and Kosko, 2001). As already mentioned, it is usually best to use triangular sets for control work as the use of other shapes simply distorts the incoming signal and adds an extra unwanted complication to the design process. The number of fuzzy sets used, and their positions, do have a major bearing on performance and this needs to be carefully considered. However, the greater the number of fuzzy sets used the greater is the computational overhead.
- (ii) The design of a fuzzy rule base is still to a large extent intuitive. Some statistical methods are available but most methods are still heuristic. Evolutionary programming techniques have shown promise in optimising rule-bases but nothing has been applied to the problem of deliberately creating non-linearities. (Wang *et al.*, 1998; Casillas *et al.*, 2005). The basic Mamdani design for the rule-base is the most difficult to tune because it is a linguistically-based system. In fact this was one of the drawbacks of the initial Mamdani approach because it does not lend itself easily to mathematical manipulation. Consequently designers tended not to trust it since they could not 'prove' that the system was stable or was always going to be stable. A major advance

was the introduction of the Sugeno, or Takagi-Sugeno-Kang, method of fuzzy inference (Sugeno, 1985). This resulted in a method which was much more mathematically and, more importantly as far as this research was concerned, much more geometrically tractable than Mamdani's.

- (iii) A powerful method of adjusting the output signal is to use an inference filter (Kiendl, 1994). This involves inserting a filter between the fuzzy output of the inference engine and the crisp output calculation stage of the controller, Figure 6.3, in practice splitting the defuzzifier into two parts. The method usually consists of applying a position-weighted function to the fuzzy output signal. The extra computational overhead in using such a filter is very small and it has the potential to provide fine-tuning for the system. To date, there have been no reports of the technique being applied to create or enhance non-linear effects.

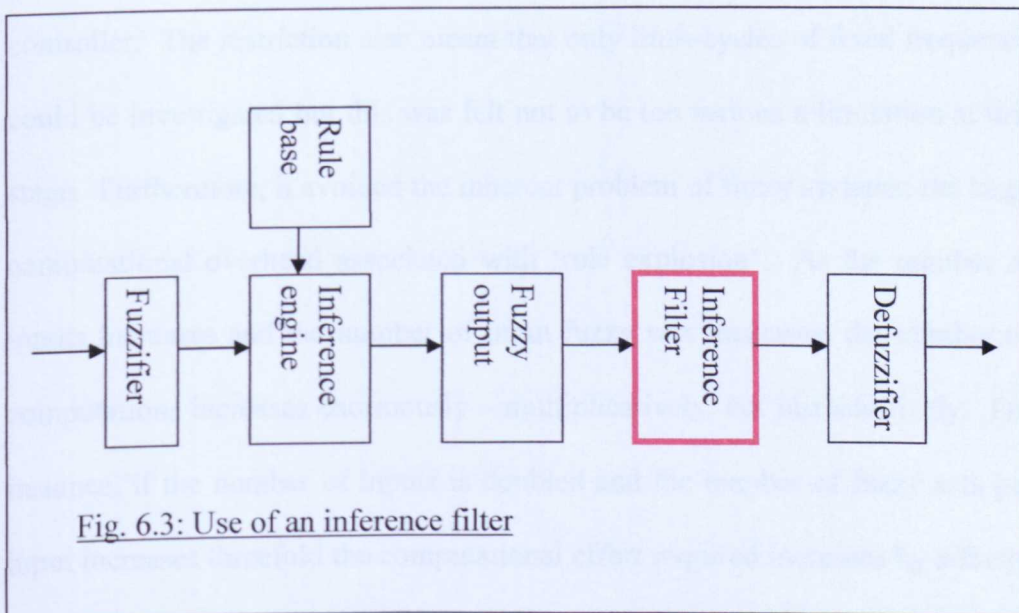


Fig. 6.3: Use of an inference filter

- (iv) If a set of circumstances necessitate the shut-down or, at least, the rapid reduction in the amplitude of the signal levels in a system (for instance,

to prevent damage) then a fuzzy controller which can handle negative rules can be used (Kiendl, 1998). This can be looked upon as a fuzzy multiplexer which would allow controller selection, not just rule selection, to be made. Although the development of a hyperinference engine is simple in principle, the Matlab fuzzy toolbox does not easily lend itself to this modification of its architecture. Because of this consideration, plus the fact that the creation of a switching mechanism is not part of the main thrust of this research, it was decided that it should be left for future development.

6.4 The tuning approach chosen

At an early stage in this investigation a decision had been made to investigate non-linearities whose describing functions only had real parts, as opposed to complex systems. This had an immediate advantage in that it simplified the fuzzy design enormously since only one crisp input will be required for each controller. The restriction also meant that only limit-cycles of fixed frequency could be investigated but this was felt not to be too serious a limitation at this stage. Furthermore, it avoided the inherent problem of fuzzy systems: the large computational overhead associated with ‘rule explosion’. As the number of inputs increases and the number of input fuzzy sets increases, the number of computations increases enormously – multiplicatively, not just additively. For instance, if the number of inputs is doubled and the number of fuzzy sets per input increases threefold the computational effort required increases by a factor of six . Actual numerical values have been included when this topic is discussed in more detail later in the thesis.

6.4.1 The fuzzy input sets

All sources (Ross, 2004; Passino *et al.*, 1997; Zimmermann, 1996) suggest that the size, number and position of the input fuzzy sets have a major bearing on performance. They affect the shape and slope of the rule-surface and the slope is directly proportional to the gain of the fuzzy controller. If narrow fuzzy sets are used then the slope of the rule-surface becomes steeper. So the gain of the fuzzy controller is dependent on the width of the fuzzy sets, particularly the central fuzzy sets. This is important because in order for a limit-cycle to occur the gain of the fuzzy controller must be greater than the gain-margin of the plant. It will be shown, in chapter seven, that when deliberately creating non-linearities the number of input fuzzy sets needs to be one more than the number of break-points.

6.4.2 The choice of inference method

As discussed earlier, the Mamdani rule-base is the most difficult to tune because it is a linguistically-based system whereas a Sugeno rule-base lends itself to mathematical manipulation. Consequently the Sugeno type was chosen as the rule-base system of choice for this investigation. In a zero-order Sugeno system a typical fuzzy rule will have the form:

*If input x is a fuzzy singleton in set A and input y is a fuzzy singleton in set B
then output $z = k$.*

where A and B are pre-defined input fuzzy sets and k is a constant. So all the output membership functions are singleton spikes. However, for the purposes

of this research the zero-order is not flexible enough. In a first-order Sugeno system the rules will be of the form:

*If input x is a fuzzy singleton in set A and input y is a fuzzy singleton in set B
then output $z = m*x+n*y+c$*

where m , n and c are constants. However, this investigation is confined to real non-linearities and therefore only a single input is required. So the rules for a first-order Sugeno system will be reduced to the form:

*If input x is a fuzzy singleton in set A then output $z = m*x+c$ (6.1)*

Higher-order Sugeno-like systems exist but they are more complicated and add nothing of value to this investigation.

6.4.3 Possible use of an inference filter

Conventional systems offer limited defuzzification methods. The two commonest in Mamdani-type systems are CoG (centre of gravity) and MoM (mean of maximum). In each separate application these options are usually interactively tested to see which produces the best results. They are ad hoc approaches and they do not enable sensitive optimization to be carried out. Sugeno types of rule-base are mathematically more tractable, particularly the first-order systems, but they still do not allow the overall range of possible fuzzy output signals to be globally adjusted.

To overcome this drawback two approaches have been developed: the inference filter approach and the torque method of defuzzification. The inference filter, Figure 6.3, is placed between the fuzzy outputs and the defuzzification stage and has the form

$$\hat{m}(u_i) = m(u_i).h(u - u_i) \dots\dots\dots (6.2)$$

Where $m(u_i)$ is the membership value of element u_i output by the inference engine and $\hat{m}(u_i)$ is the modified membership value of the same element after passing through the inference filter. The magnitude of each individual fuzzy output is modified by some function h which is distance selectable from some pivotal position u .

By contrast, in the torque approach an extra term $\sum_{k=1}^r m_k(u).(u - u_k)$ is added to the formula for the crisp output. It can be looked upon as the torque induced by $m_k(u)$ about some pivotal point u . $m_k(u)$ is the degree of activation of rule k and $(u - u_k)$ is the distance of the fuzzy output singleton from the pivot position. Although the notions of inference filter and torque were developed separately, the torque approach is simply a subset of the inference filter method. MATLAB does allow a limited form of inference filtering in that it allows each rule to be selectively weighted. However, they can only be weighted by a factor between 0 and 1. Use was made of this limited form in chapter eight when selective design of partial inverse describing functions was investigated.

6.5 The fuzzy approach used in this research

The Sugeno type 1 design approach was chosen, using single input signals and triangular input fuzzy sets which were coincident but not overlapping. It was

found that this approach enabled discontinuous non-linearities which have straight sections between break points to be easily designed. Further it was found that the technique could be extended to continuous non-linearities. The author is not aware of any other technique which allows non-linear systems to be designed in such an easy and straightforward manner. The technique has allowed a sufficiently large range of non-linearities to be rapidly developed for it to be possible to identify patterns between them which were not previously obvious and had not been reported in the literature. The effect of critical points in influencing the range of input values that can exist before a limit-cycle is activated, the interaction between adjacent limit-cycles if the system possesses more than one, when is a system going to be stable and when unstable – all points which will be considered in the next two chapters.

6.6 A Template

In order that fuzzy logic Sugeno type one functions could be easily designed, a template was devised in which all the important features of each design could be seen at a glance. An example of the template is shown in Figure 6.4 and the complete details can be found in Appendix A3.5.

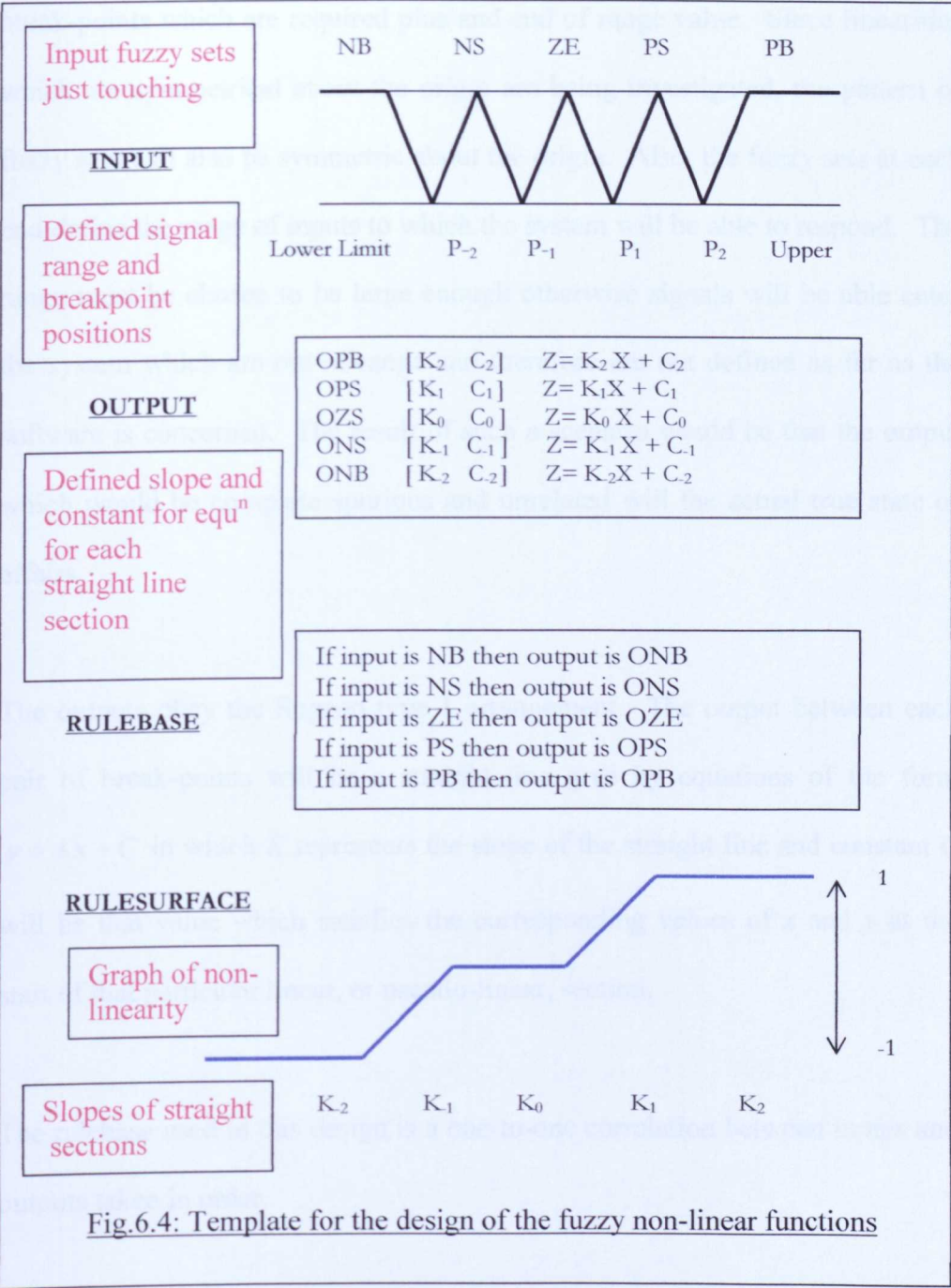


Fig.6.4: Template for the design of the fuzzy non-linear functions

The template starts with the triangular fuzzy, type one, triangular inputs which are just touching. However, care has to be taken to ensure that there is no ambiguity of output if the input coincides with a nullinsection point. In some cases to avoid this problem very slight overlapping of the input fuzzy sets was allowed. The number of inputs required is determined by the number of

break-points which are required plus and end of range value. Since linearities which are symmetrical about the origin are being investigated, the pattern of fuzzy sets will also be symmetric about the origin. Also, the fuzzy sets at each end define the range of inputs to which the system will be able to respond. The range must be chosen to be large enough otherwise signals will be able enter the system which are out-of-range and therefore are not defined as far as the software is concerned. The result of such a scenario would be that the output which would be complete spurious and unrelated will the actual true state of affairs.

The outputs obey the Sugeno type 1 arrangement. The output between each pair of break-points will be a straight line give by equations of the form $y = Kx + C$ in which K represents the slope of the straight line and constant C will be that value which satisfies the corresponding values of x and y at the start of that particular linear, or pseudo-linear, section.

The rulebase used in this design is a one-to-one correlation between inputs and outputs taken in order.

Finally, the template shows the shape of the fuzzy rule-surface which, with this design arrangement, should correlate exactly with the shape of the nonlinearity which would be seen if a unit ramp were input to this designed module in an open-loop arrangement.

6.7 Conclusion

This chapter started with an overview of where fuzzy logic came from. The next item was a brief introduction to fuzzy logic theory as applied to control systems. An explanation of the various tuning methods available was presented, including some modern methods not often seen but which have a potential relevance to this research. After giving an overview of the tuning methods available decisions were made about which were most appropriate for this work. Some of the methods which have the potential to be useful but are not directly relevant at this stage have been earmarked for future work. In general, the techniques which have been so earmarked suffer from excessive computational time requirements rather than any inherent complexity in their implementation.

Once the decisions had been made about the fuzzy techniques to be used the next step was to apply them to the design of non-linear systems. This work is reported in chapter seven, together with an analysis of their performance.

Chapter Seven

SIMULATION AND ANALYSIS OF FUZZY NON-LINEAR SYSTEMS

7.1 Introduction

As discussed in chapter six, the first-order Sugeno type of fuzzy system was found to be the most suitable for this research since the inference rules take the form of linear equations, as stated in equation (6.1):

*If input x is a fuzzy singleton in set A then output $z=m*x+c$*

(Note: that the symbol K was used to represent the slope when deriving the basic describing function. Although the symbols m and K are completely interchangeable, to avoid confusion the symbol K has been used in the rest of this work)

It was also decided that triangular input sets would be used since, as already reported, the shape of fuzzy sets is not critical for control system and fuzzy membership values for that shape are the most easily calculated and do not introduce any distortions or biases into the fuzzy calculations.

A series of trials took place with different numbers of input sets, with varied widths, with sets which overlapped by differing amounts and with sets which just touched but did not overlap. It was found that the shape of the fuzzy input sets did not really matter if they were just touching, except for square-shaped sets which caused considerable cross-talk at their junctions. However, no drawbacks were found using triangular sets, with trapezoidal end sets, so the research continued with these in use.

With the input sets just touching each other, the membership functions of the fuzzy input quantities appeared to play no part in the output of the inference section. However, if there was any overlap of the input sets then the membership functions did make a contribution and this was seen in the smoother transition from one section to another. Because it was easy to see, and to design, the output responses, the subsequent research was conducted with the sets just touching. When any particular non-linearity was designed, although the output response could be seen by looking at the rule-surface (which was two-dimensional in this case where there was only one input) the subsequent model was tested in the SIMULINK arrangement shown in Figure 7.1, this was to ensure that some unexpected distortion(s) had not crept in.

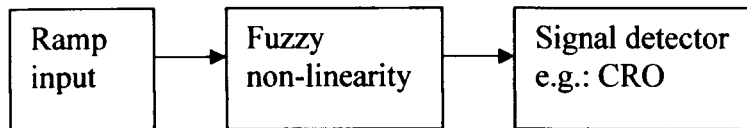


Fig. 7.1: Testing arrangement to ensure the non-linearities have the correct parameters.

For every fuzzy input a corresponding output of the form $z=Kx+c$ was designed and the rule-base consisted of a one-to-one relation between the inputs and the outputs. Using this technique, replicas of the standard modules from chapter five were designed and tested to evaluate how closely the fuzzy methods could emulate ordinary classical results. When this had been successfully demonstrated the techniques were used to design the more

involved non-linearities whose behaviour had been predicted in chapter five but which are not ordinarily seen.

7.2 Fuzzy Hard Saturation

The fuzzy non-linearity consists of three fuzzy input sets and three outputs with the values shown in Figure 7.2. This was embedded in the general SIMULINK circuit shown in Figure 7.3. From this arrangement the set of signals shown in Figure 7.4 was obtained.

From Figure 7.4, the measured frequency of oscillation was 2.40 ± 0.03 rad/s. This compares favourably with the calculated frequency of 2.45 ± 0.001 rad/s (see Appendix 2.1). From Figure 7.4, the measured magnitude of the limit-cycle was 2.05 ± 0.03 which compared with the calculated magnitude of oscillation, from Figure 5.5, of 2.15 ± 0.02 . So, in Figure 7.4, comparing with the results shown in Figure 5.3, the fuzzy simulation was a close fit to the saturation module supplied with SIMULINK package. The results from Figure 5.3 were superimposed in Figure 7.4 for comparison.

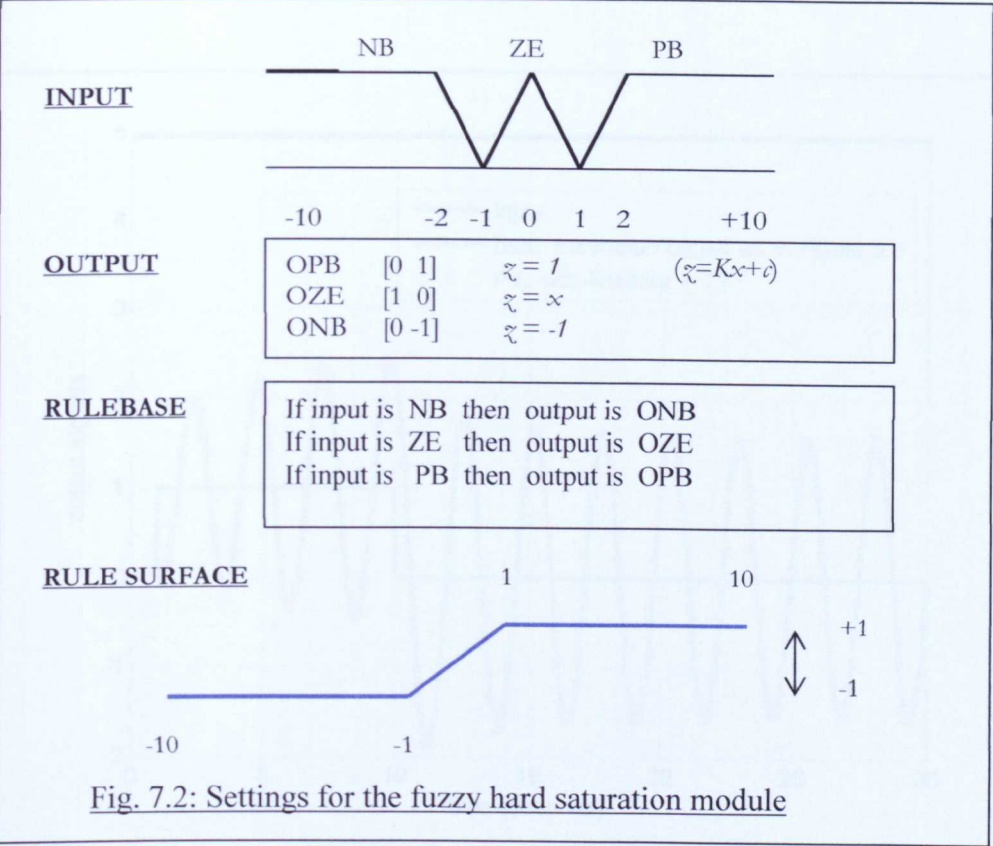


Fig. 7.2: Settings for the fuzzy hard saturation module

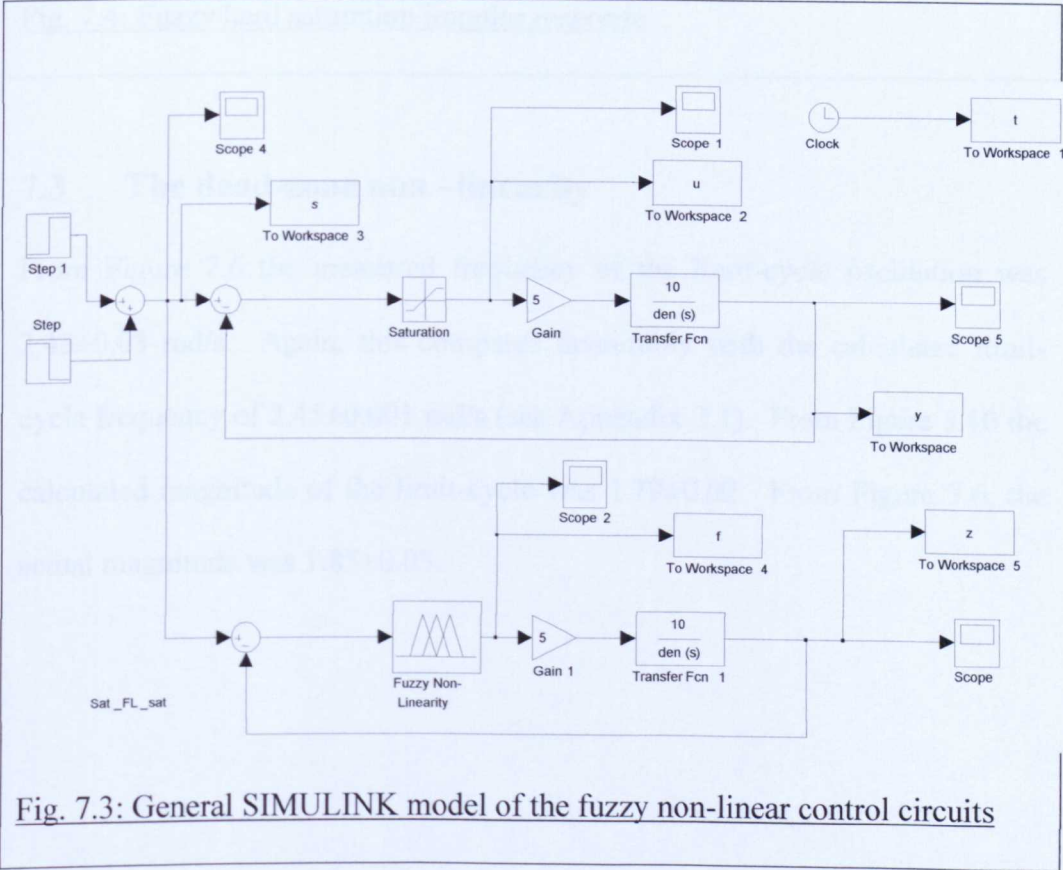


Fig. 7.3: General SIMULINK model of the fuzzy non-linear control circuits

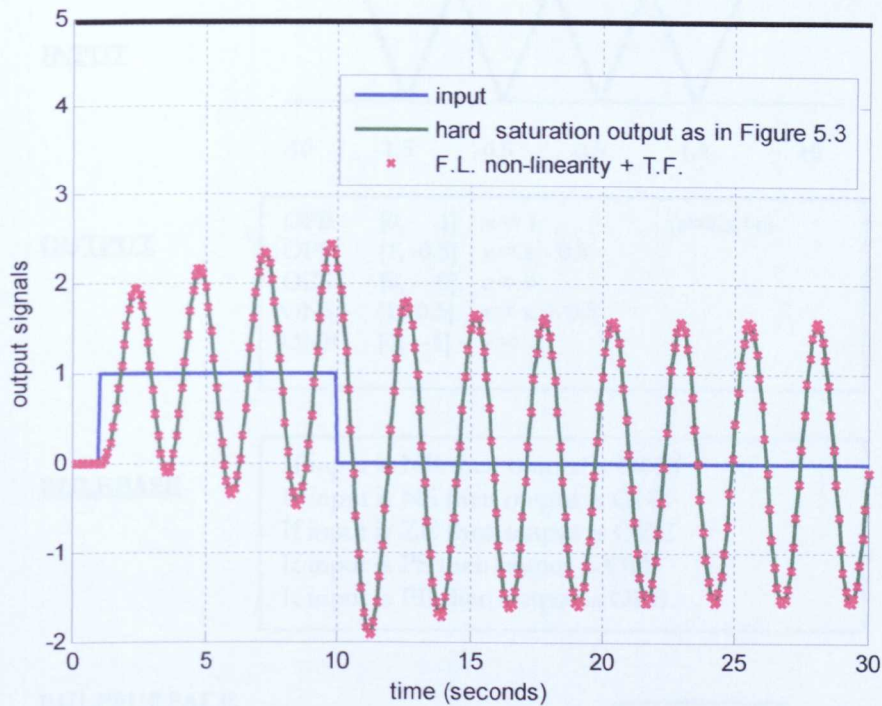


Fig. 7.4: Fuzzy hard saturation impulse response

7.3 The dead-zone non –linearity

From Figure 7.6 the measured frequency of the limit-cycle oscillation was 2.43 ± 0.03 rad/s. Again, this compares favourably with the calculated limit-cycle frequency of 2.45 ± 0.001 rad/s (see Appendix 2.1). From Figure 5.10 the calculated magnitude of the limit-cycle was 1.79 ± 0.02 . From Figure 7.6, the actual magnitude was 1.85 ± 0.03 .

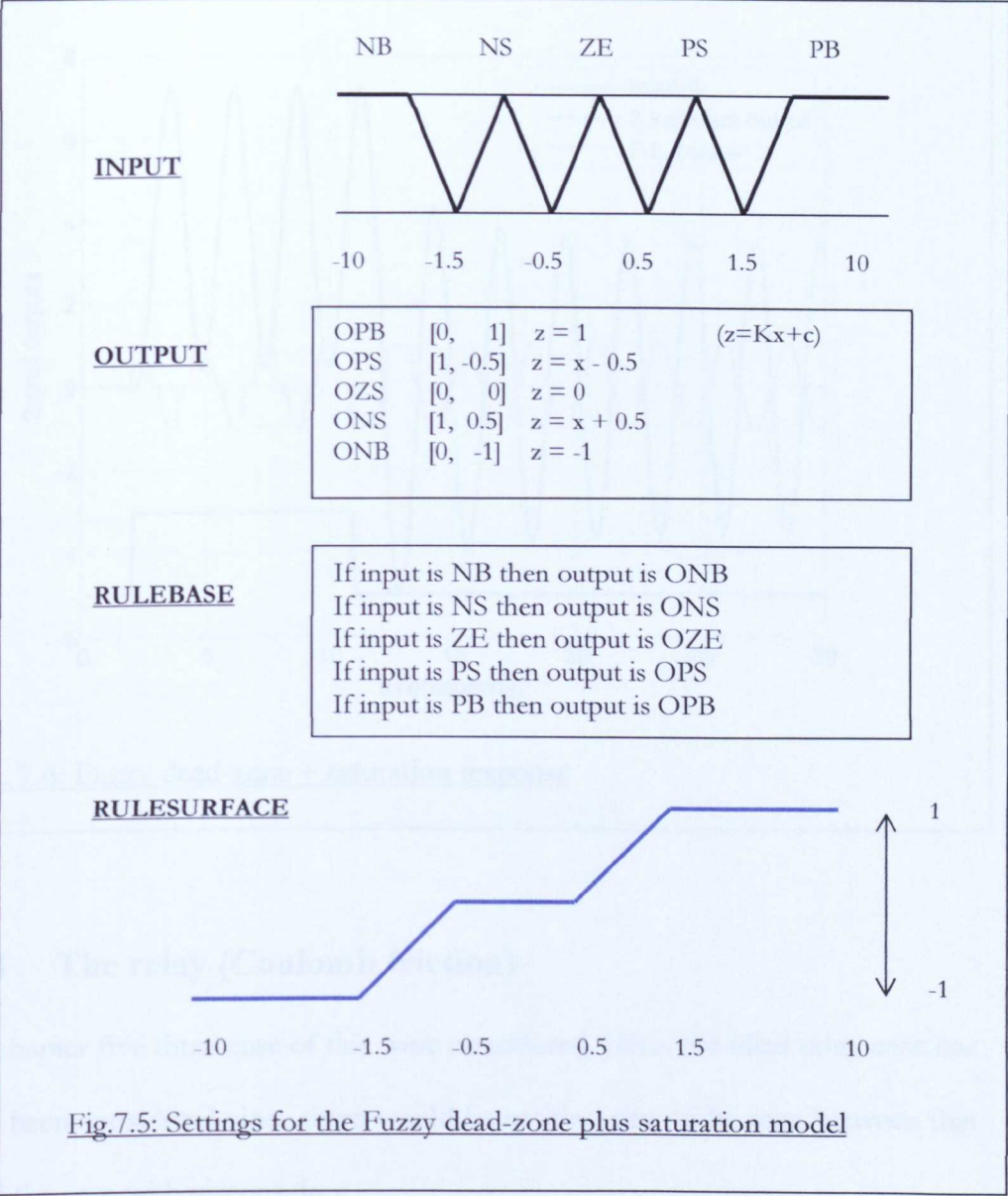


Fig.7.5: Settings for the Fuzzy dead-zone plus saturation model

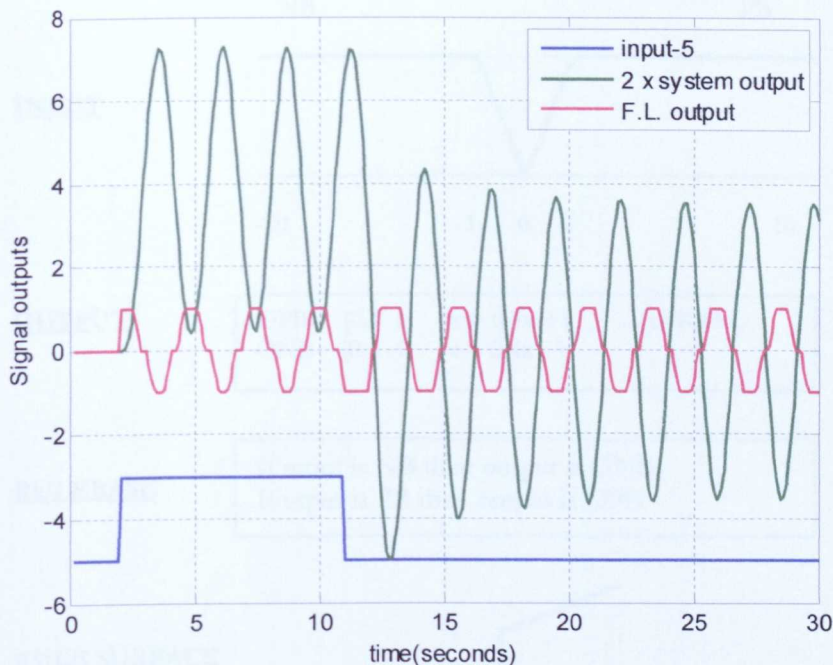


Fig. 7.6: Fuzzy dead-zone + saturation response

7.4 The relay (Coulomb friction)

In chapter five three case of this were considered. Here, the ideal relay case has not been considered since there would be nothing new to be seen between that and the case with viscous drag.

7.4.1 Coulomb friction plus viscous drag

From Figure 7.8 the measured frequency of the limit-cycle oscillation was 2.42 ± 0.03 rad/s. From appendix 2.1 the calculated limit-cycle frequency was 2.45 ± 0.001 rad/s. From Figure 5.17 the calculated magnitude of the limit-cycle was 7.48 ± 0.02 and from Figure 7.8, the actual magnitude was 7.68 ± 0.03 .

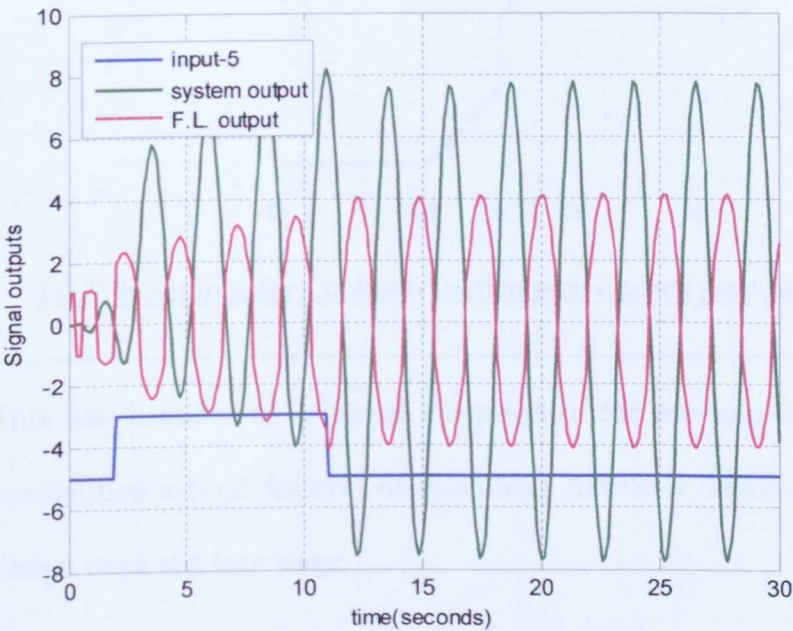
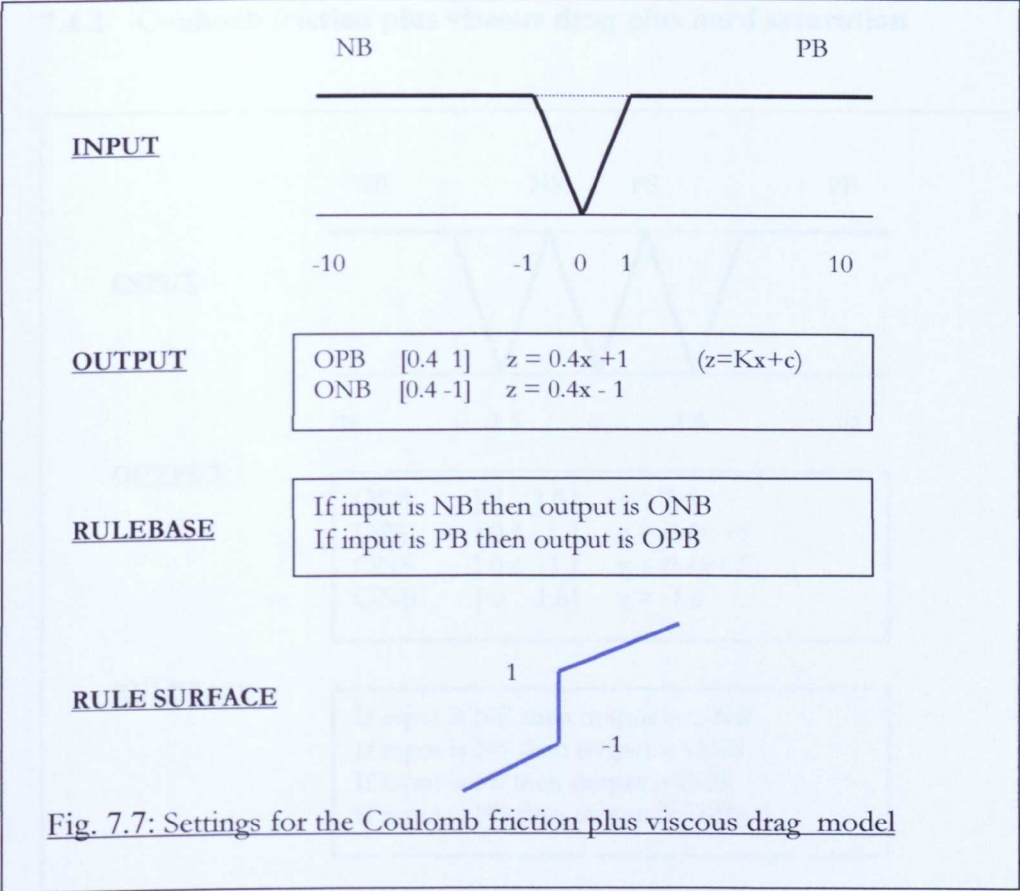


Fig. 7.8: Coulomb friction (Relay) plus viscous drag response

7.4.2 Coulomb friction plus viscous drag plus hard saturation

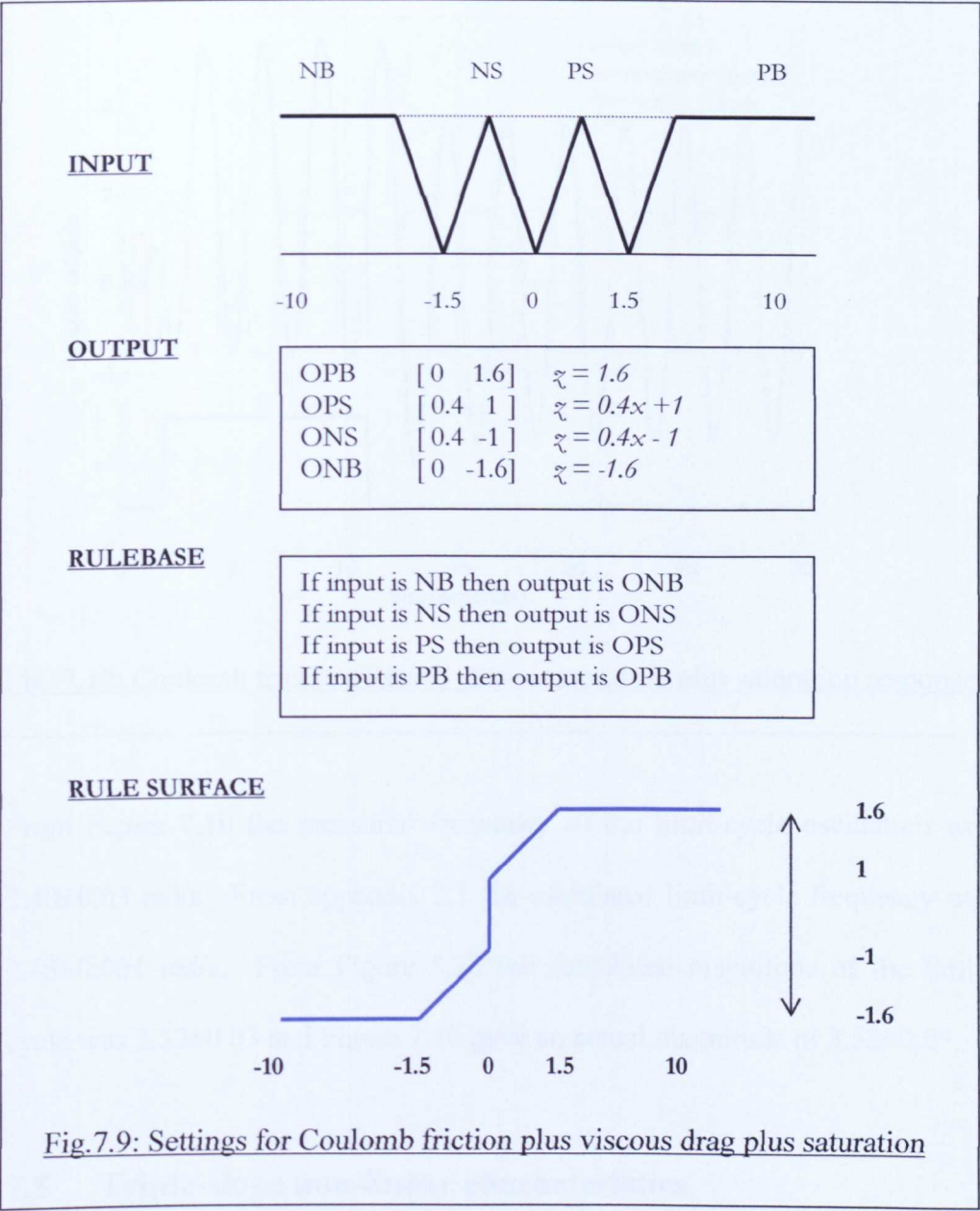


Fig.7.9: Settings for Coulomb friction plus viscous drag plus saturation

This was included as a logical extension of the previous section because it exemplified several features of describing functions which could be used in design work at a later stage.

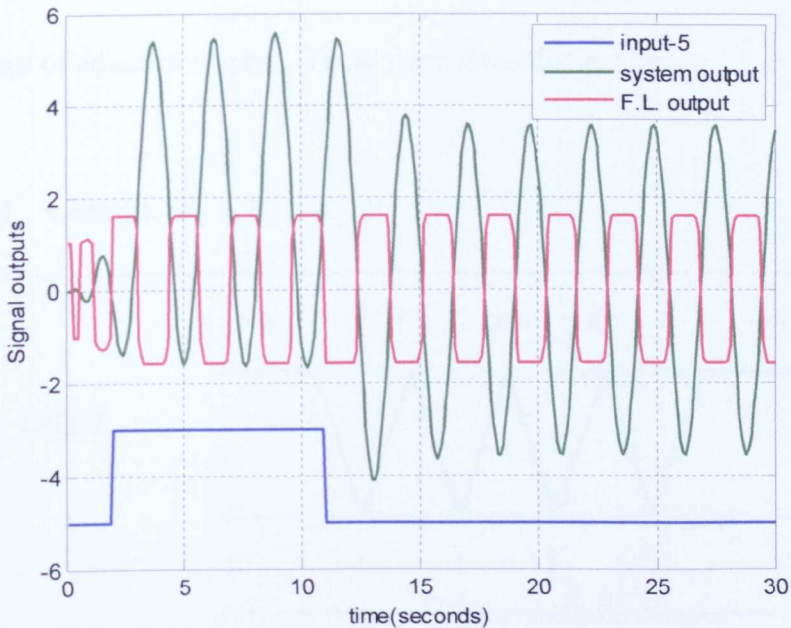


Fig. 7.10: Coulomb friction (Relay) plus viscous drag plus saturation response

From Figure 7.10 the measured frequency of the limit-cycle oscillation was 2.40 ± 0.03 rad/s. From appendix 2.1 the calculated limit-cycle frequency was 2.45 ± 0.001 rad/s. From Figure 5.20 the calculated magnitude of the limit-cycle was 3.52 ± 0.03 and Figure 7.10 gave an actual magnitude of 3.53 ± 0.03 .

7.5 Triple-slope non-linear characteristics

Starting with this group of non-linearities the design technique that had been developed began to show its worth. These non-linearities had much more involved describing functions which exemplified a rich variety of non-linear features. As mentioned in chapter four, there were three main cases to be considered for the triple slope non-linearity and these depended on the relative

values of adjacent slopes. This was because the formulae for the describing function depended on the relationship $(K_{i-1} - K_i)$ where K_{i-1} & K_i were the values of adjacent slopes. There were three distinct cases:

7.5.1 Case (i): $K_0 > K_1 > K_2$

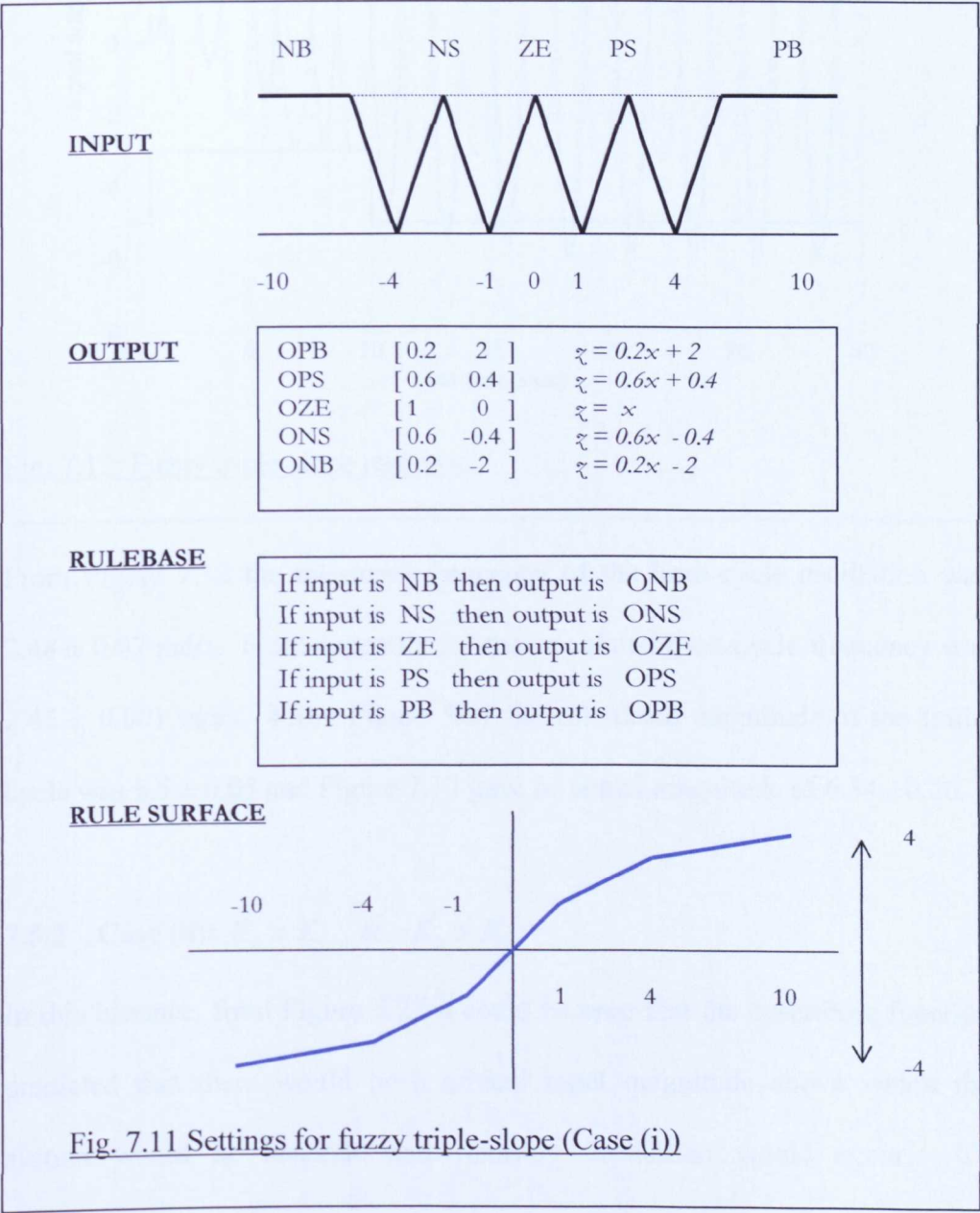


Fig. 7.11 Settings for fuzzy triple-slope (Case (i))

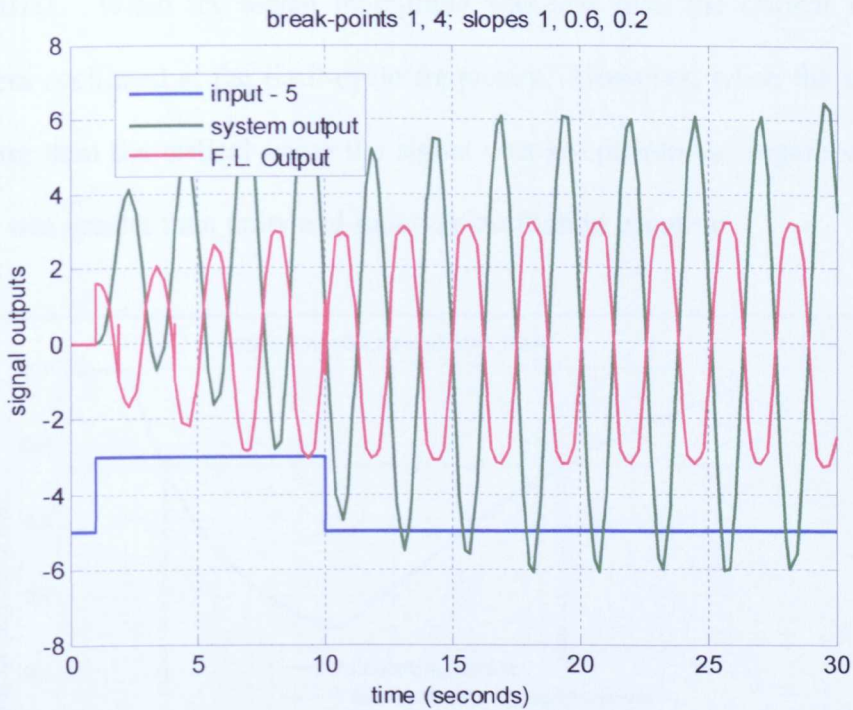


Fig. 7.12: Fuzzy triple-slope response

From Figure 7.12 the measured frequency of the limit-cycle oscillation was 2.48 ± 0.07 rad/s. From appendix 2.1 the calculated limit-cycle frequency was 2.45 ± 0.001 rad/s. From Figure 5.21 the calculated magnitude of the limit-cycle was 6.5 ± 0.05 and Figure 7.12 gave an actual magnitude of 6.34 ± 0.26 .

7.5.2 Case (ii): $K_0 > K_1$ & $K_2 > K_1$

In this instance, from Figure 5.22 it could be seen that the describing function predicted that there would be a critical input magnitude above which the system would be unstable and runaway oscillation would occur. To demonstrate this effect, an annotated version of Figure 5.22 is given here (now Figure 7.13).

When gain = 50, the critical value = 7.5 ± 0.3 and the magnitude of limit-cycle = 1.6 ± 0.02 . When the signal magnitude was less than the critical value the system oscillated at the limit-cycle frequency. However, when the input was greater than the critical value the signal was swept into the region where the gain was greater than unity and runaway oscillation occurred.

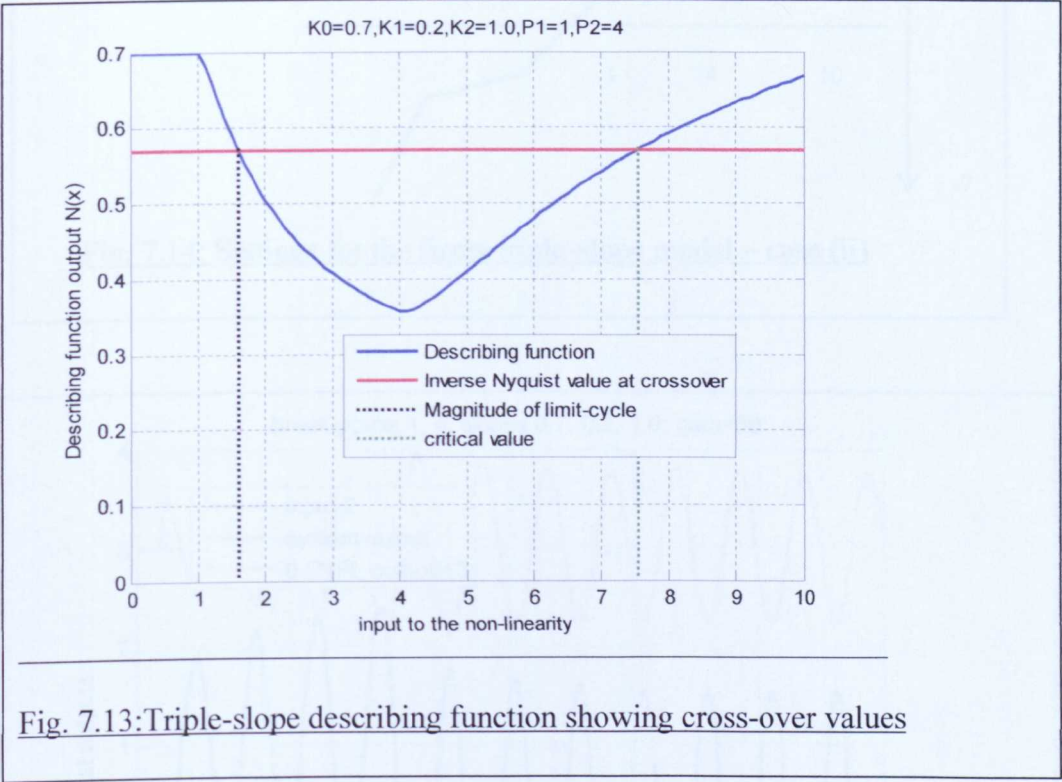
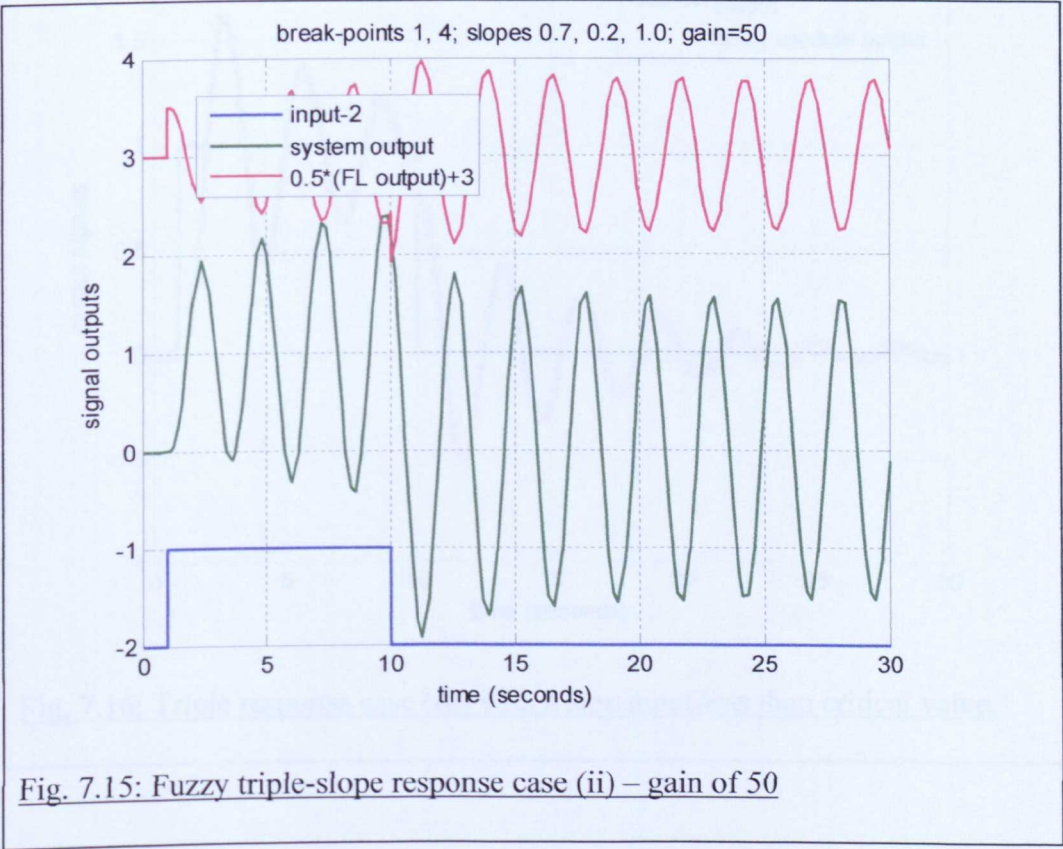
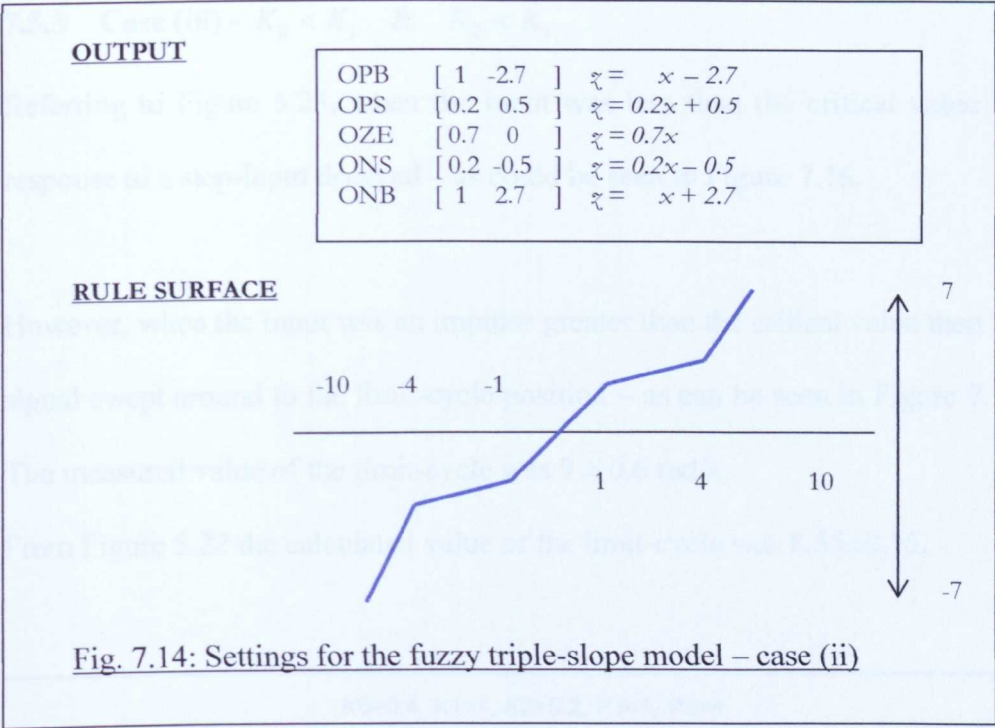


Fig. 7.13:Triple-slope describing function showing cross-over values



7.5.3 Case (iii) - $K_0 < K_1$ & $K_2 < K_1$

Referring to Figure 5.23, when the input was less than the critical value the response to a step-input decayed – as could be seen in Figure 7.16.

However, when the input was an impulse greater than the critical value then the signal swept around to the limit-cycle position – as can be seen in Figure 7.17.

The measured value of the limit-cycle was 9 ± 0.6 rad/s.

From Figure 5.22 the calculated value of the limit-cycle was 8.55 ± 0.15 .

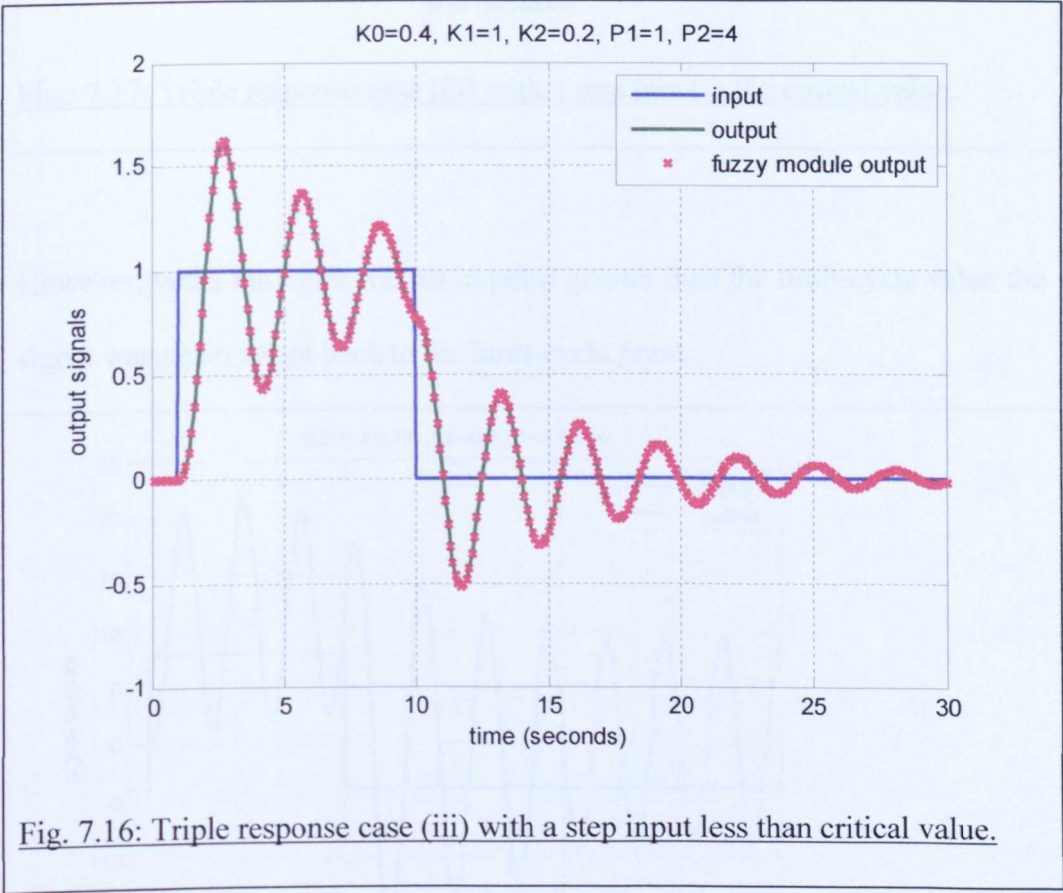
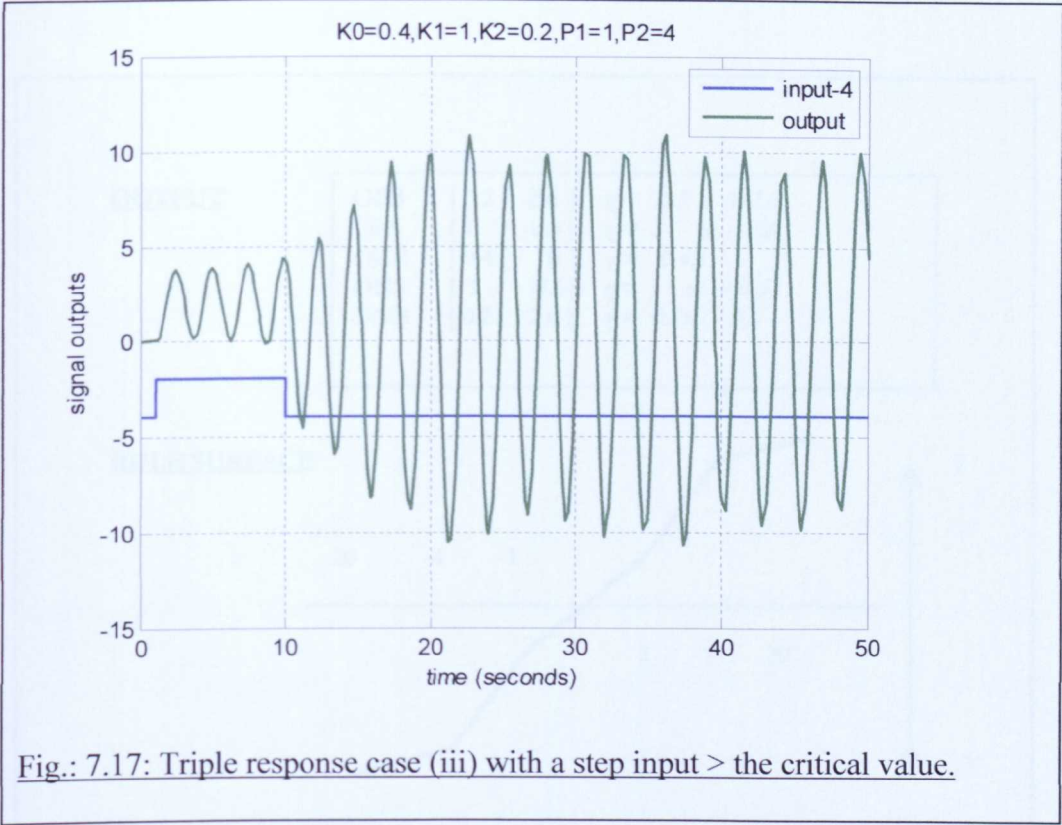
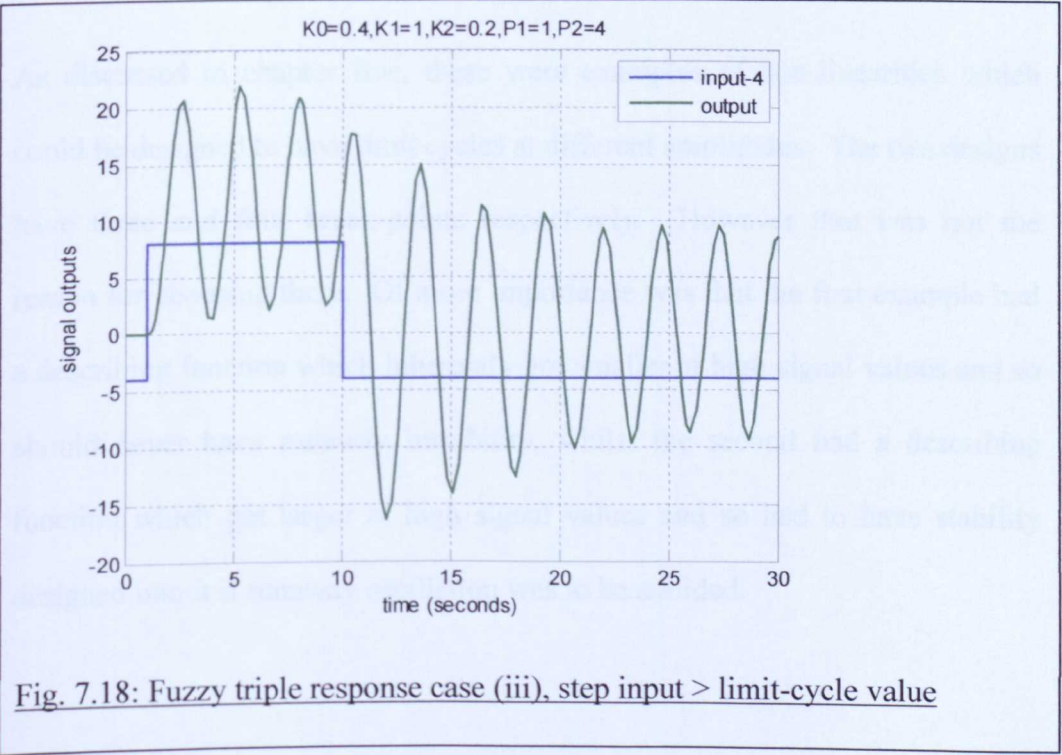


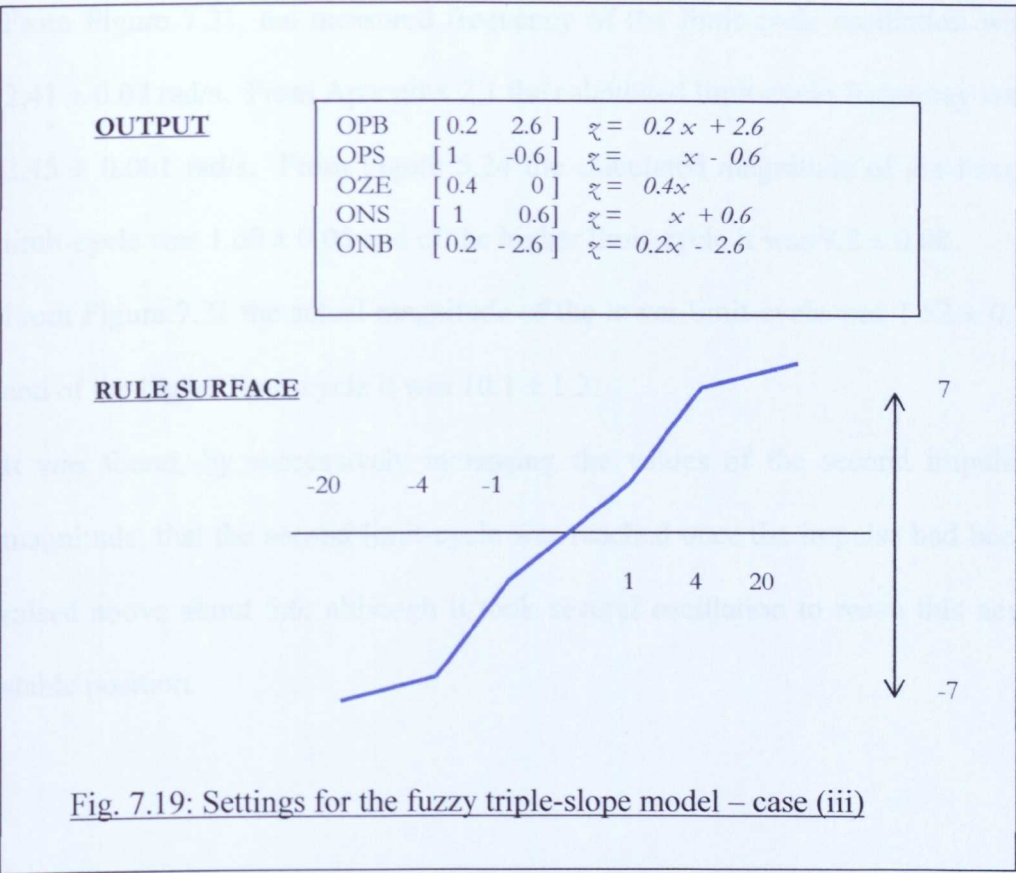
Fig. 7.16: Triple response case (iii) with a step input less than critical value.



However, when the input was an impulse greater than the limit-cycle value the signal was again swept back to the limit-cycle point.



7.6.1 The three break-point case



7.6 Non-linearities which produce more than one limit-cycle

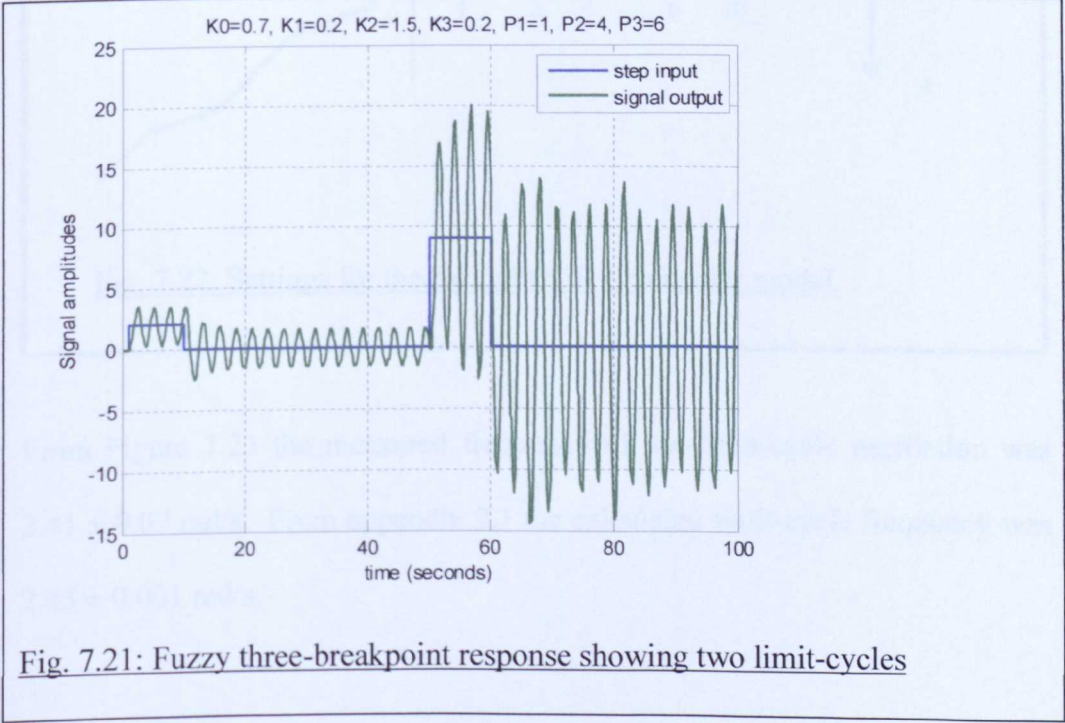
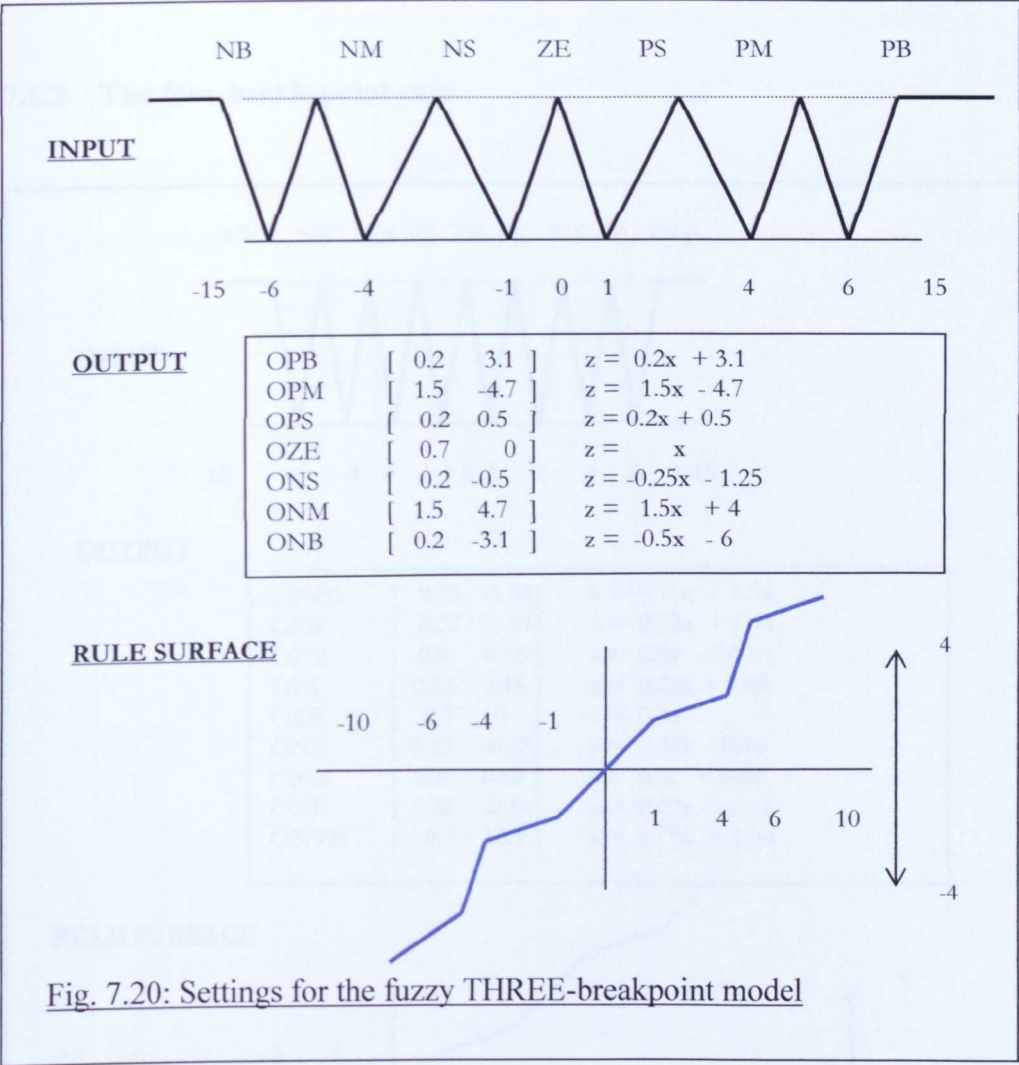
As discussed in chapter five, these were examples of non-linearities which could be designed to have limit cycles at different amplitudes. The two designs have three and four break-points respectively. However that was not the reason for choosing them. Of more importance was that the first example had a describing function which inherently got smaller at high signal values and so should never have runaway instability, whilst the second had a describing function which got larger at high signal values and so had to have stability designed into it if runaway oscillation was to be avoided.

7.6.1 The three break-point case

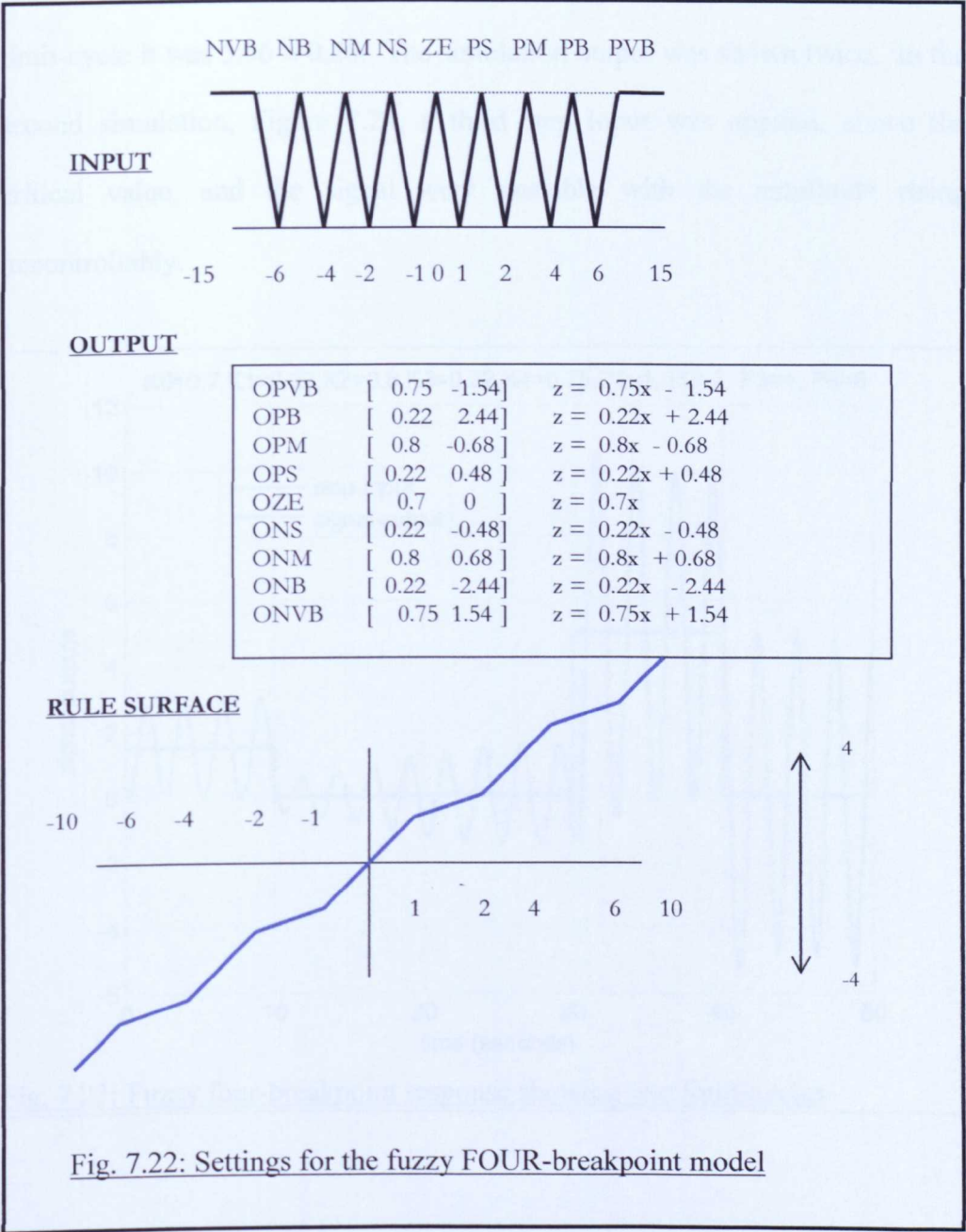
From Figure 7.21, the measured frequency of the limit-cycle oscillation was 2.41 ± 0.07 rad/s. From Appendix 2.1 the calculated limit-cycle frequency was 2.45 ± 0.001 rad/s. From Figure 5.24 the calculated magnitude of the lower limit-cycle was 1.60 ± 0.05 and of the higher limit-cycle it was 9.2 ± 0.08 .

From Figure 7.21 the actual magnitude of the lower limit-cycle was 1.52 ± 0.7 and of the higher limit-cycle it was 10.1 ± 1.2 .

It was found, by successively increasing the values of the second impulse magnitude, that the second limit-cycle was reached once the impulse had been raised above about 5.6; although it took several oscillation to reach this new stable position.



7.6.2 The four break-point case



From Figure 7.23 the measured frequency of the limit-cycle oscillation was 2.41 ± 0.07 rad/s. From appendix 2.1 the calculated limit-cycle frequency was 2.45 ± 0.001 rad/s.

From Figure 5.23 the calculated magnitude of the lower limit-cycle was 1.8 ± 0.05 and of the higher limit-cycle it was 5.50 ± 0.05 . From Figure 7.23 the actual magnitude of the lower limit-cycle was 1.85 ± 0.26 and of the higher limit-cycle it was 5.56 ± 0.26 . The simulation output was shown twice. In the second simulation, Figure 7.24, a third step input was applied, above the critical value, and the signal went unstable with the amplitude rising uncontrollably.

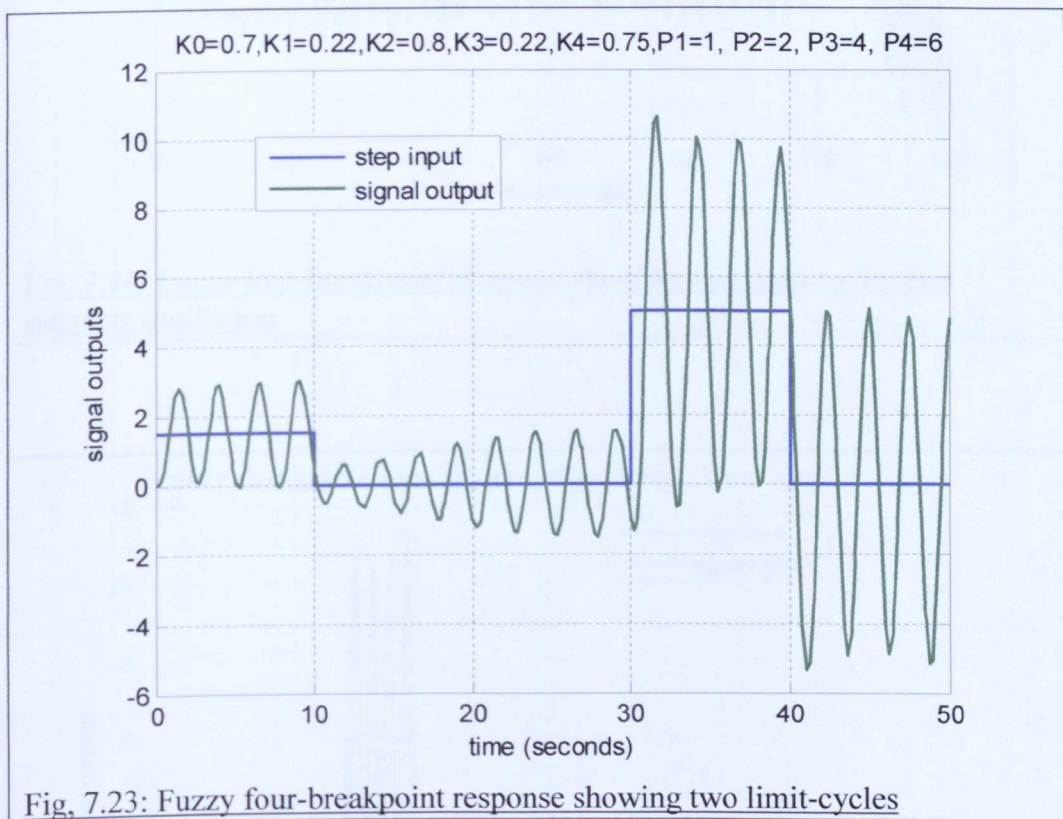
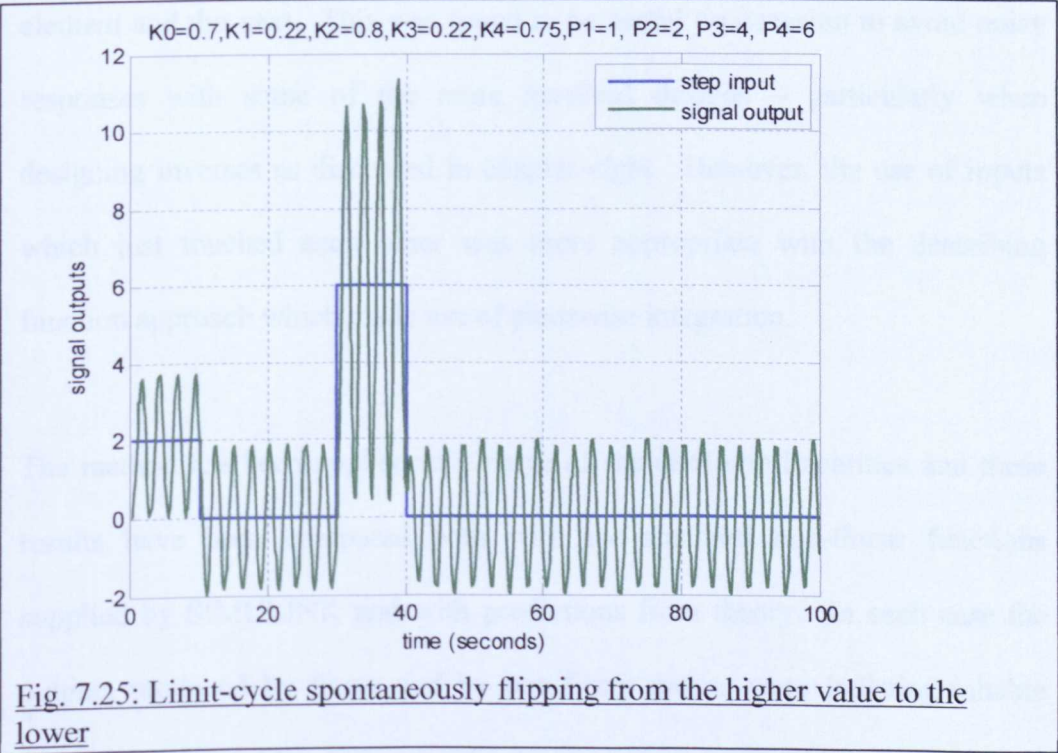
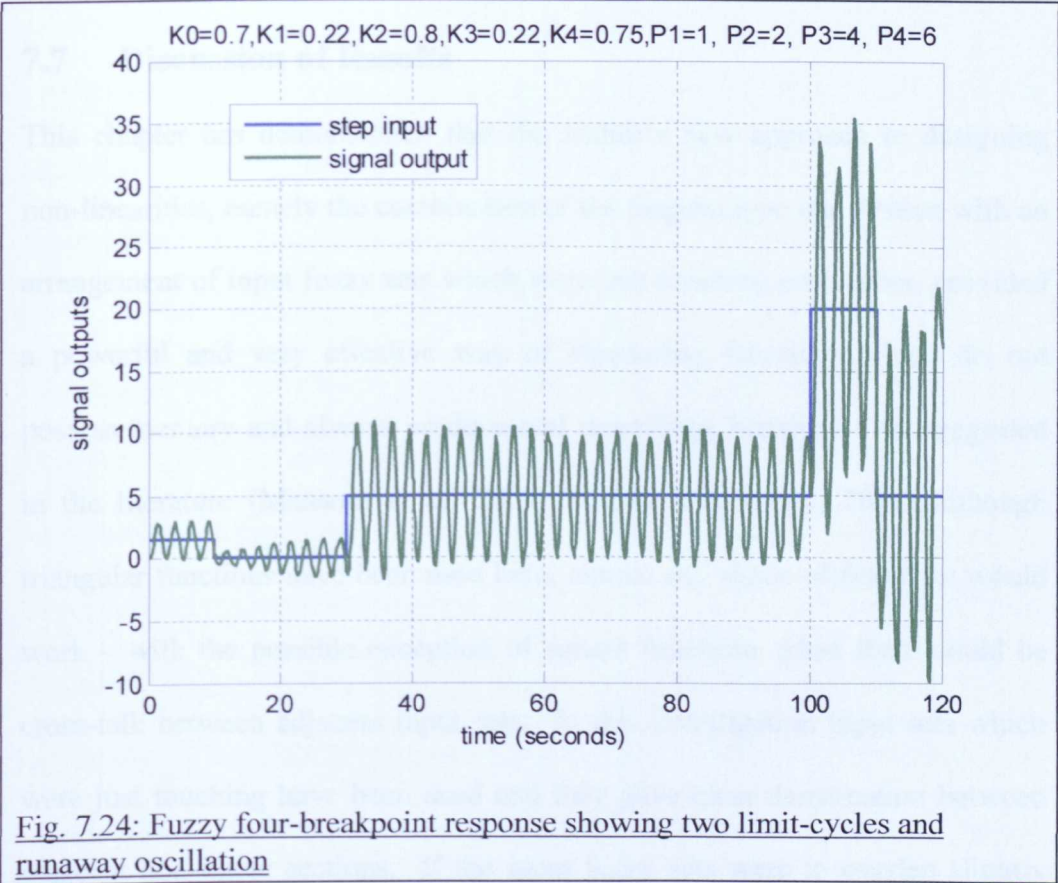


Fig. 7.23: Fuzzy four-breakpoint response showing two limit-cycles



7.7 Discussion of Results

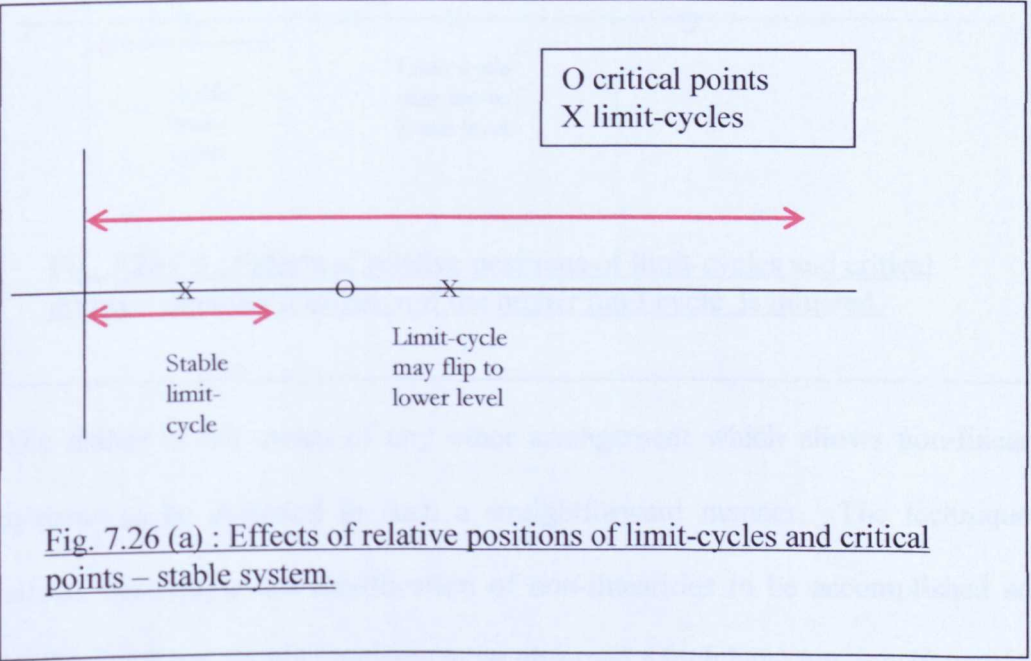
This chapter has demonstrated that the author's new approach to designing non-linearities, namely the combination of the Sugeno type one system with an arrangement of input fuzzy sets which were just touching each other, provided a powerful and very effective way of simulating functions which do not possess memory and always produce real describing functions. As suggested in the literature (Mansoor *et al.* 2007; Mitiam and Kosko, 2001) although triangular functions have been used here, almost any shape of fuzzy set would work – with the possible exception of square functions when there could be cross-talk between adjacent input sets. In this investigation input sets which were just touching have been used and they gave clear demarcation between adjacent non-linear sections. If the input fuzzy sets were to overlap slightly then the output functions would show smooth transitions from one non-linear element and the next. This was found to be useful on occasion to avoid noisy responses with some of the more involved designs – particularly when designing inverses as discussed in chapter eight. However, the use of inputs which just touched each other was more appropriate with the describing function approach which made use of piecewise integration.

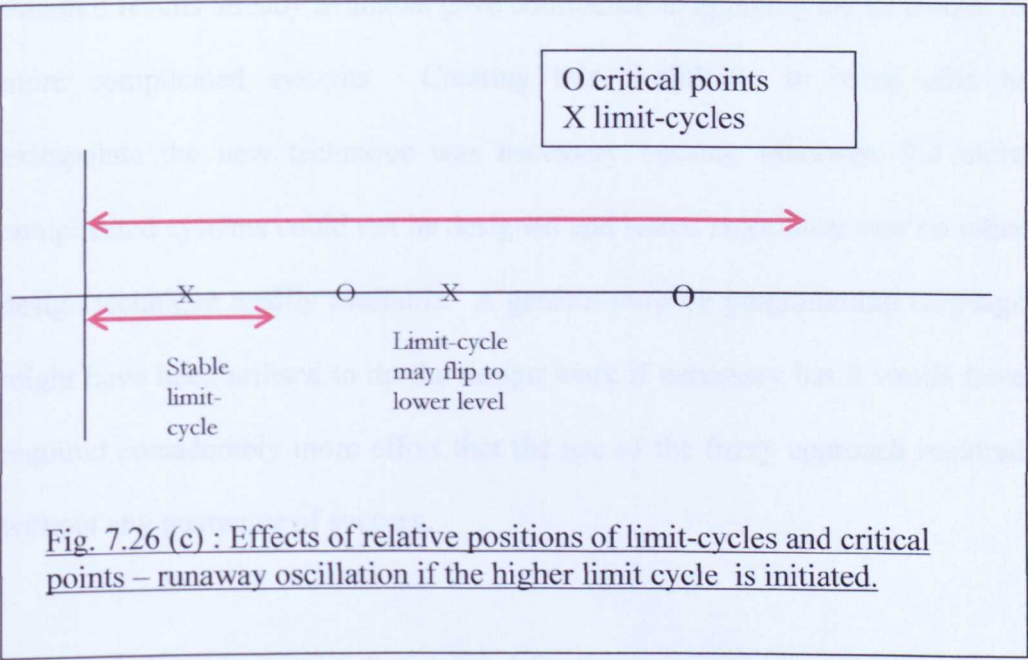
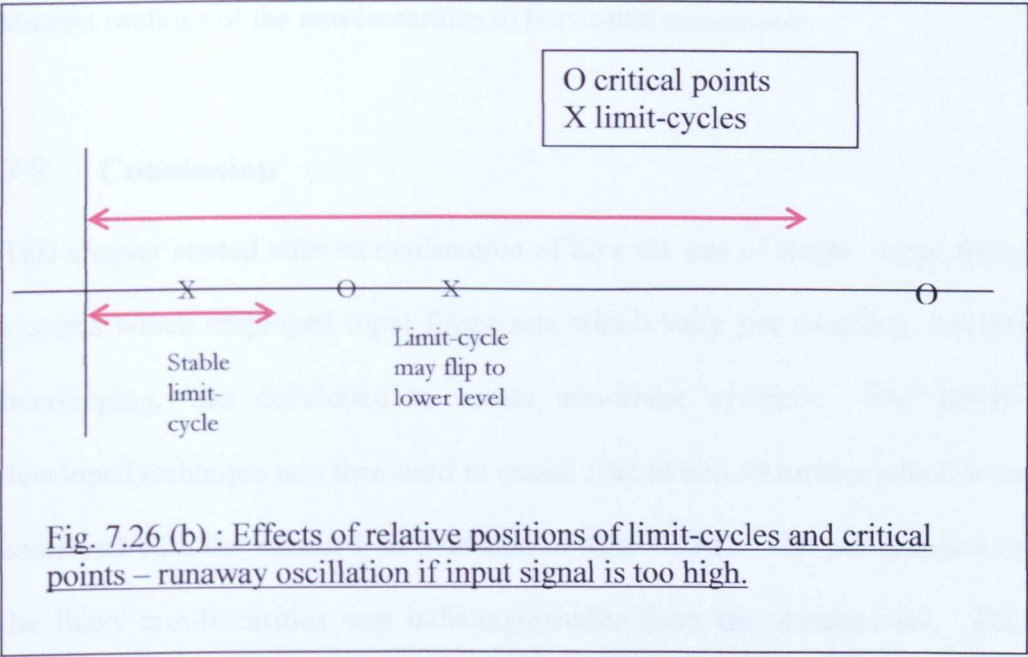
The method has been applied to a range of standard non-linearities and these results have been compared both with the standard non-linear functions supplied by SIMULINK and with predictions from theory. In each case the outputs produced by fuzzy and by non-fuzzy means were indistinguishable from each other. Also, they agreed with theoretical prediction to within one or two percent.

The success of the technique, plus its very simplicity, gave confidence in extending it to more sophisticated structures, in particular structures which had the potential to exhibit more than one limit-cycle. Two examples have been given: the three break-point case and the four break-point case. In both cases two limit cycles were predicted and were found to exist with the magnitudes expected. From Figures 5.24 and 5.25, showing describing functions with the inverse Nyquist superimposed on them, some further behaviour-patterns could be predicted associated with the cross-over points where the magnitudes of the describing function were rising. These were positions, labelled as critical points, where the describing functions were going from regions in which the overall gain was less than unity to places where the gain was greater than unity. When these were encountered, the magnitudes of the signals, as they went around the feedback loop, were going to be swept to ever-higher values and might, if there was a downward cross-over point higher up, be swept to that limit-cycle position, otherwise instability would occur. This behaviour-pattern could be found over the whole range of signal-magnitudes, leading to situations where (i) non-linear systems were stable for low magnitudes but not at high (e.g.: dead-zone, and the four-breakpoint case), (ii) were unstable at low magnitudes but then became stable at high (e.g.: the triple-slope characteristic, the four-breakpoint characteristic and Coulomb friction), or (iii) have regions of stability or of instability (e.g.: triple-slope case (ii), the three and four breakpoint cases).

When a limit-cycle oscillation occurred, the magnitude of the signal was increasing and decreasing in a cyclical fashion which means that it was moving

up and down the describing function characteristic. If the magnitude of the oscillation was greater than the distance between the limit-cycle position and the next critical point then there was a chance that the signal would jump out of the stable oscillation position and either enter the region of instability or move to the next limit-cycle location. This effect was more likely to occur with the higher limit-cycle because its magnitude of oscillation was greater. An example of this effect was demonstrated in Figure 7.25. In Figure 7.24, the start of runaway oscillation was shown. It was possible to design the non-linearity such that the higher critical point did not exist or was much higher than the oscillation of the higher limit-cycle. It was this feature: that the design of new and exotic non-linearities was now possible, which enabled the investigation of non-linear signal cancellation and modification. These results were summarized in Figure 7.26 (a, b and c). The red lines indicate the range of oscillation associated with each limit-cycle.





The author is not aware of any other arrangement which allows non-linear systems to be designed in such a straightforward manner. The technique allows the design and modification of non-linearities to be accomplished so easily that it has enabled patterns to be observed which have previously passed

unnoticed – such as the fact that describing functions transform the slopes of straight sections of the non-linearities to horizontal asymptotes.

7.8 Conclusion

This chapter started with an explanation of how the use of single –input fuzzy systems which employed input fuzzy sets which were just touching, but not overlapping, was developed to create non-linear systems. The newly-developed technique was then used to create a set of non-linearities which were compared with the standard set available in SIMULINK. The performance of the fuzzy non-linearities was indistinguishable from the standard set. This agreement between the fuzzy method for creating non-linearities and the standard results already available gave confidence in applying the technique to more complicated systems. Creating this confidence in being able to extrapolate the new technique was necessary because otherwise the more complicated systems could not be designed and tested since there was no other design technique readily available. A general-purpose programming language might have been utilised to do the design work if necessary but it would have required considerably more effort than the use of the fuzzy approach required without any guarantee of success.

Fuzzy non-linearities were then created for the more complicated systems for which describing functions had been developed in chapter five. Using these systems the behaviour of multiple limit-cycles as tested, and the relative stability and the interactions between them, was both predicted and observed.

The concept of critical points was introduced, with their relevance being both demonstrated and discussed.

A feature which has emerged is that a describing function transforms the slopes (gains) of the linear sections of the non-linearities into horizontal asymptotes on the diagram used for displaying the describing function locus use in this research, Fig. 3.5. Furthermore, the break points appear as vertical asymptotes on the same type of diagram. This immediately produces as technique for quickly creating look-up tables for non-linearities by using these asymptotic effects and avoiding the use of mathematical calculations.

The next stage, in chapter eight, was to use these techniques both to create 'inverse' non-linearities, which could cancel-out the effects of standard non-linearities, and to adjust signals so that non-linear effects which existed could be modified at will. But this would first require an investigation into what effect an 'inverse' function, in the sense employed in this research, would have on a signal.

Chapter Eight

Using Fuzzy Logic to Modify and Control Linear Systems

8.1 Introduction

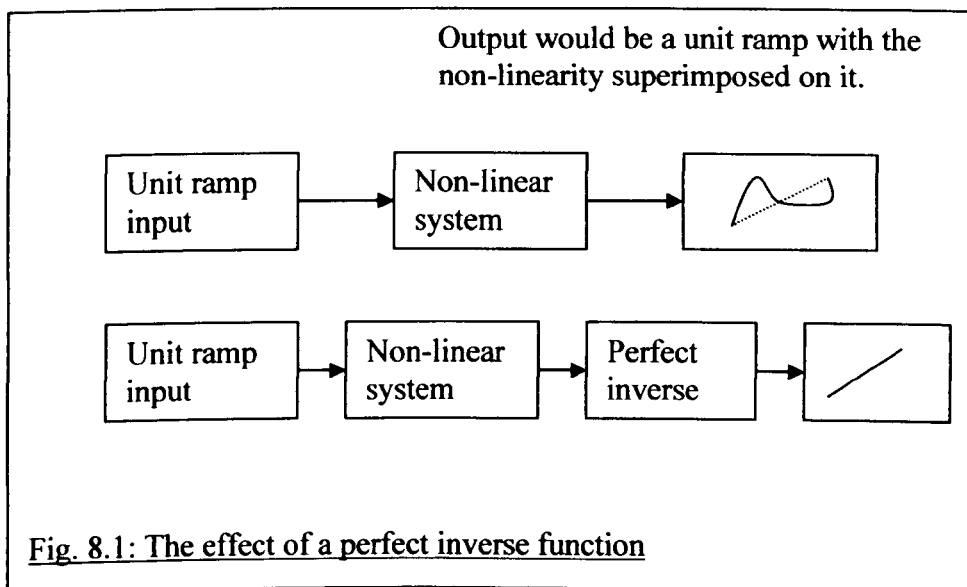
The received wisdom was that it was inherently impossible to develop a perfect inverse of a non-linear system to cancel out all non-linear behaviour, Lee *et al.* (2001), Vashkov *et al.* (1998), Foo *et al.* (2001), because the principle of superposition did not apply. However, although it might be true that it was not possible to find inverses analytically there were cases in which complete inverses could be found by geometric means. In particular, it was possible to find inverses for systems which have real describing functions, as has been demonstrated in this chapter. In any case, often a complete inverse function did not have to be found: all that was necessary was that the shape of the native describing function became modified sufficiently for it to no longer cross the inverse Nyquist locus.

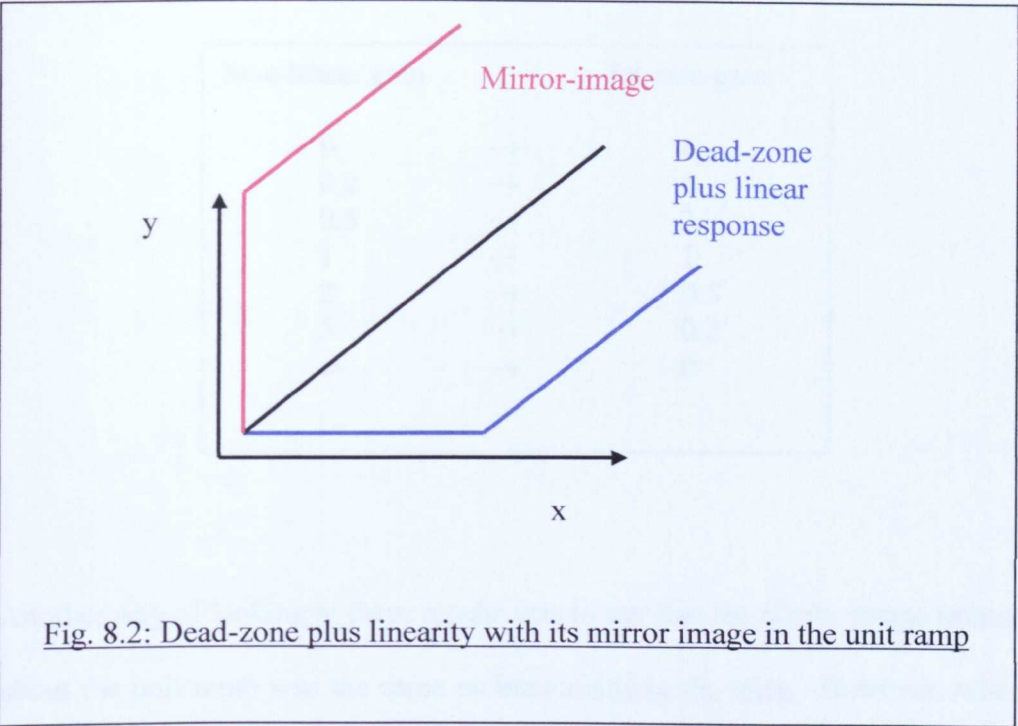
In this chapter the concept of an inverse function, as applied in this investigation, was considered, some conclusions drawn, and a suitable approach developed for creating such functions. The approach was then tested by applying it to the standard functions already developed. The ‘inverses’ of the standard functions thus created were then placed in series, in open loop, with the original functions and the results of applying ramp inputs were investigated. The research proceeded with the investigation of ‘partial inverse’ functions which removed selected sections of a non-linearity. Finally, the

problem of creating inverses of continuous non-linearities was considered and the initial method was extended to include this group.

8.2 The effect of an inverse function

If a ramp input was applied to systems with an embedded non-linearity in an open-loop circuit then the output signal would be representative of that non-linearity. If a perfect inverse of the non-linearity was then placed in series the output should simply have been the original ramp input, as in Figure 8.1. What became evident during the course of this research was that it was indeed possible, provided that the nonlinearity had a real describing function, to create a perfect inverse. Furthermore, such inverses were simply the mirror images of the original nonlinearities about the unit ramp. This was because all the inverse had to do was to nullify the instantaneous gain introduced by the non-linearity at each point along its path. If this effect was looked at more closely some unexpected results emerge. In Figure 8.2 the effect of dead-zone, followed by a unit linear response, was shown together with its mirror-image about a unit ramp.





What was immediately obvious was that the mirror image was the same as the non-linear characteristic for Coulomb friction plus a linearity. So Coulomb friction, or the ideal relay, was the inverse of dead-zone and vice-versa. This result could also be predicted algebraically as shown in Appendix A3.2. The general transformations as far as the slopes (gains) of the piecewise linear sections of the non-linearities were concerned were shown in Table 8.1.

Table 8.1: General scaling transformations to create inverses

Non-linear gain		Inverse gain
0	→	∞
0.2	→	5
0.5	→	2
1	→	1
2	→	0.5
5	→	0.2
∞	→	0

Another way of looking at these results was to say that the mirror image rotated about the unit ramp was the same as interchanging the axes. However, when creating the fuzzy inverse a few subtleties had to be taken into account. When dealing with non-linearities which were approximated by a series of straight lines it was a simply enough task to construct the inverse by hand but if the inverse of a continuous non-linear function was to be constructed, or if the linear approximation was at all complicated, then it was easier to produce a two-column table of data in which one column represented the magnitude of the input signal and the other the corresponding values of the function. To create the inverse, the two columns were simply reversed. However, when actually constructing the fuzzy inverse function then the break-points for the inverse were the projections onto the y-axis with reference to the original non-linearity; whereas they were given by the projections onto the x-axis for the original function. Since they were the projections of the original function onto the y-axis these new break-points would not be symmetrical about the 45° line when compared with the original function's break-points.. Likewise the slopes

would be with reference to the y-axis from the original non-linearity rather than the x-axis. Since these two axes were orthogonal, if K^* was the mirror image of the original slope K , it followed that (See Appendix A3.1)

$$K.K^* = 1 \dots\dots\dots (8.1)$$

It ws then possible to develop a general method for designing a fuzzy non-linearity, and its inverse, for systems with real describing functions.

8.3 The design of a fuzzy non-linearity and its inverse

In all cases a Sugeno type 1 fuzzy controller design was used. Again, as shown in the template in chapter six, triangular fuzzy input sets which just touch each other were the shape of choice. Just touching, so that the adjacent slopes were continuous and also so that adjacent slopes of the non-linearity have their own exclusive fuzzy sets without overlap or undefined sections.

(In both the initial function and its inverse the fuzzy inputs used were isosceles triangles. Although, as mentioned later, the requirement on the shape of the fuzzy sets was not too prescriptive.)

The outputs were of the form $y = Kx + c$ where K was the slope of the relevant section of the non-linearity and, in each case, c was a suitable constant to ensure that the adjacent slopes of the outputs were connected.

The breakpoints, P_i , were the values of input x at which the gradients of the various straight-line slopes which approximate the non-linearity changed value.

Since the function passed through the origin it followed that $P_0 = 0$. Also, K_i was the slope between break-points P_i and P_{i+1} .

c_i was the constant in the output section of the fuzzy non-linearity, with $c_0 = 0$

Since the values of c_i had to be chosen so that adjacent slopes of the outputs were connected (just touching) it followed that the constant of the n^{th} straight line was given by:

$$c_n = \sum_{i=1}^n (K_{i-1} - K_i) P_i \dots\dots\dots (8.2)$$

Just as the triangular inputs of the starting functions were projections of the non-linearity onto the x-axis, so the triangular inputs of its inverse function were projections of the same non-linearity onto the y-axis. It was shown, in equation (8.1), that $K.K^* = 1$ where K was the gradient of a section of the non-linearity and K^* was the gradient of the corresponding section of its inverse. Consequently, from Figure 8.3, the gradients of the inverses were:

$$K_i^* = 1 / K_i \quad \text{for all values of } i \quad \dots\dots\dots (8.3)$$

The breakpoints on the inverse function were given by P_i^* , with $P_0^* = 0$.

It followed, from Figure 8.3, that:

$$P_n^* = \sum_{i=1}^{n-1} (K_{i-1}^* - K_i^*) P_i + K_{n-1}^* P_n \dots\dots\dots (8.4)$$

The values c_i^* represented the constants of the output section of the fuzzy inverse non-linearity and $c_0^* = 0$ since the inverse function would also pass through the origin. They also had to be chosen so that adjacent slopes of the outputs were connected (just touching) so it followed that the constant of the n^{th} straight line would be given by:

$$c_n^* = \sum_{i=1}^n (K_{i-1}^* - K_i^*) P_i^* \dots\dots\dots (8.5)$$

The derivations of these equations are shown in Appendix 3.3.

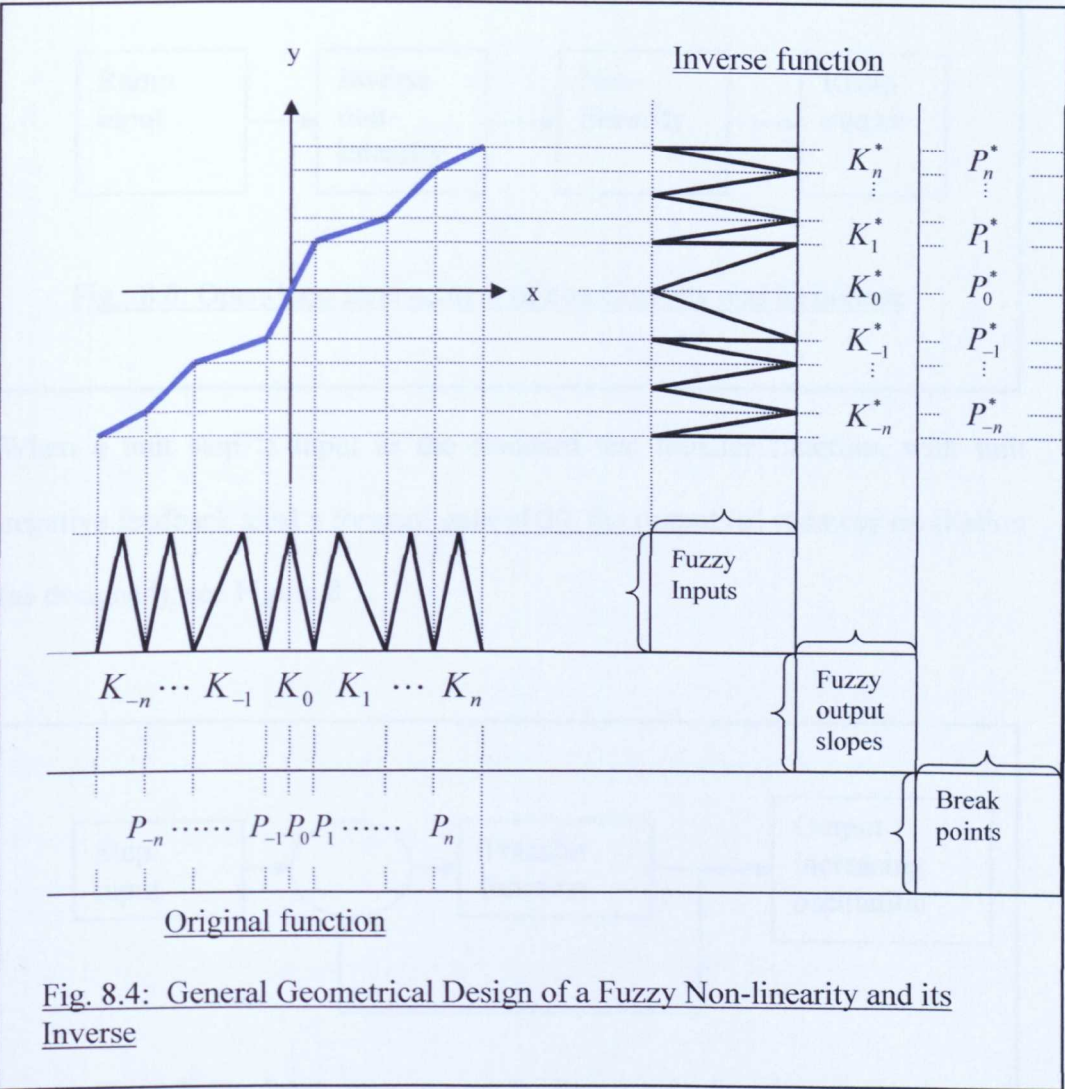
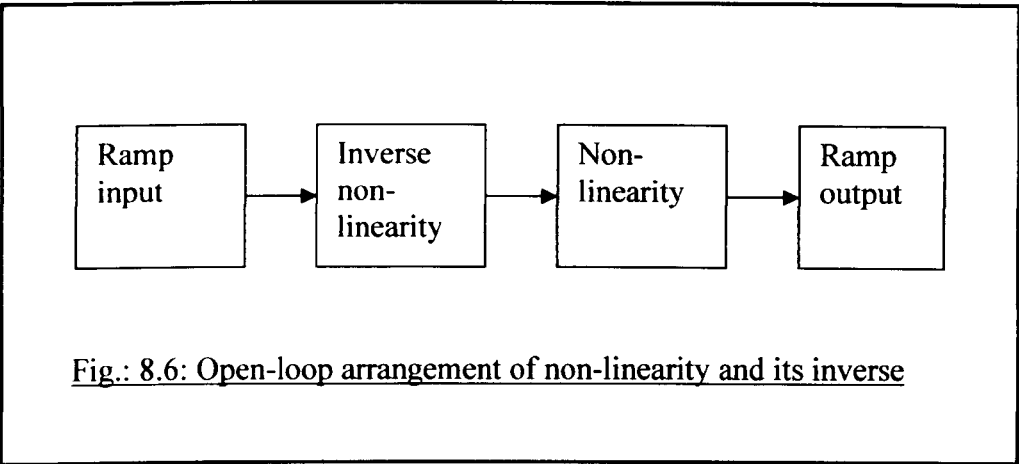


Fig. 8.4: General Geometrical Design of a Fuzzy Non-linearity and its Inverse

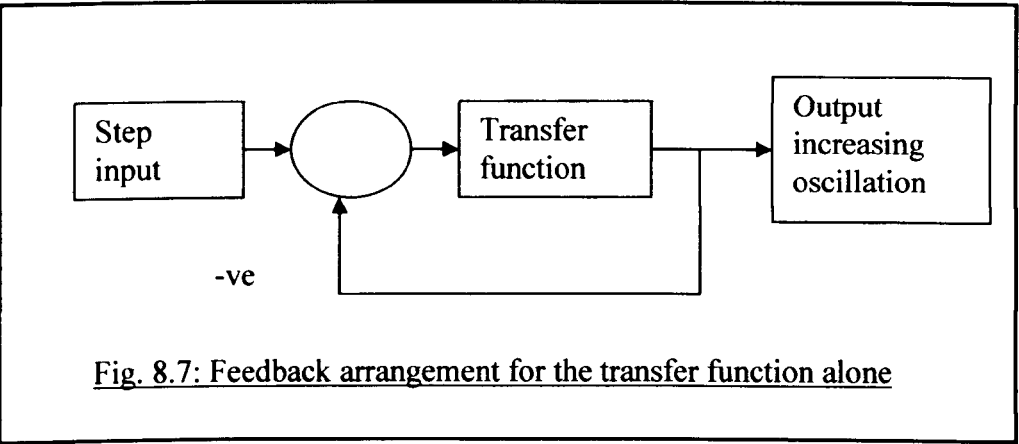
8.4 Proving the method

8.4.1 A block-diagram explanation of how the design method was tested.

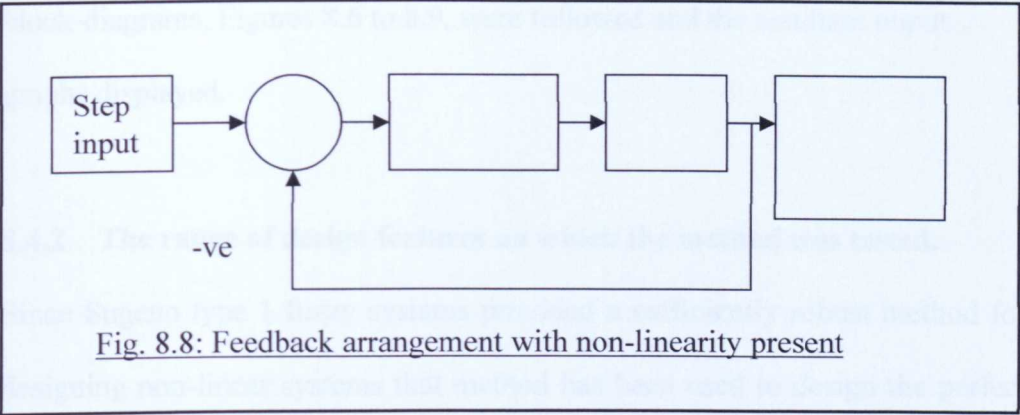
To recapitulate., when a ramp input is fed into an open-loop arrangement of a non-linearity in series with another non-linearity which is its perfect inverse the output will be a ramp of the same slope as the input, Figure 8.6.



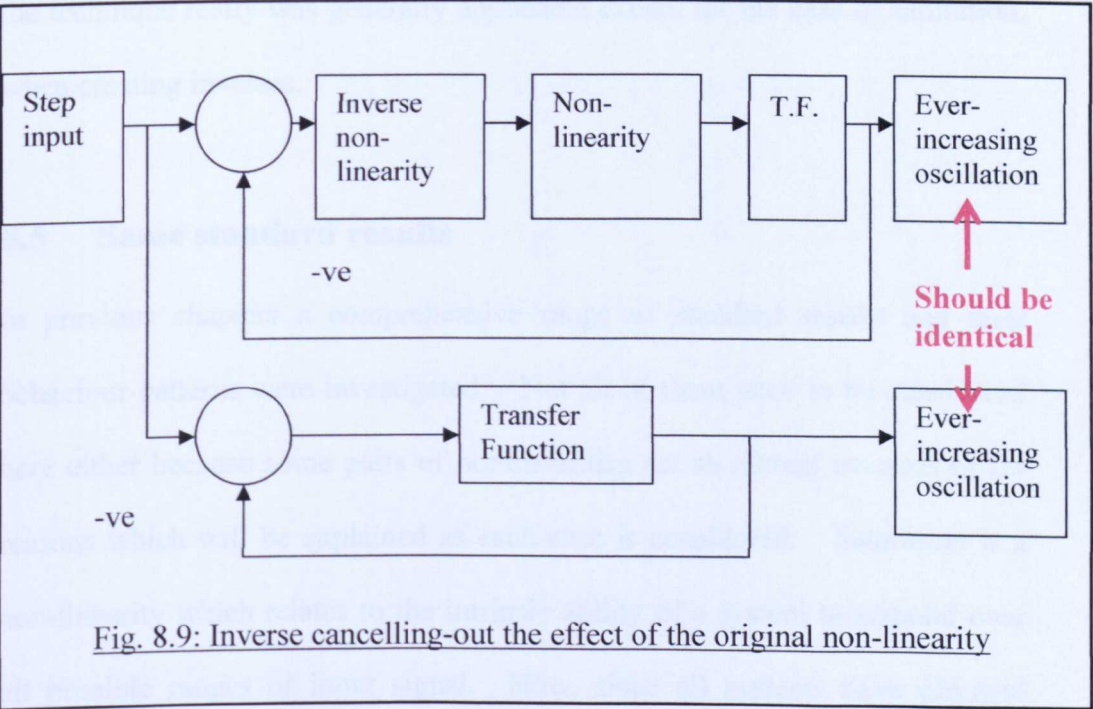
When a unit step is input to the standard test transfer function, with unit negative feedback and a forward gain of 50, the output is runaway oscillation (as designed), see Figure 8.7.



When the test non-linearity is placed in series with the transfer function in the forward path of the closed loop, and a step input is applied, the output will be a limit-cycle (as designed), see Figure 8.8.



Finally, Figure 8.9 shows the situation when a perfect inverse of the non-linearity is placed in series with the non-linearity and the transfer function in the forward path of the closed loop, and a step input is applied. The output will be ever-increasing oscillation. If this output is compared with the resultant behaviour when the transfer function was on its own in the forward path of the closed loop the two runaway oscillatory outputs should be identical – if the inverse is a perfect inverse of the original non-linearity.



With each inverse non-linearity under test, the routines outlined in the above block-diagrams, Figures 8.6 to 8.9, were followed and the resultant output graphs displayed.

8.4.2 The range of design features on which the method was tested.

Since Sugeno type 1 fuzzy systems provided a sufficiently robust method for designing non-linear systems that method has been used to design the perfect inverses of the standard non-linearities already encountered and these have then been implemented in fuzzy form and their effectiveness evaluated. The approach was then used for some of the systems which possessed multiple limit-cycles in order to check that the technique really was a robust design tool. Further, it was shown to be possible to use the design tool to modify non-linearities in more selective ways rather than just eliminating them, particularly when used in conjunction with the describing function/inverse Nyquist graphs. Finally, two examples of continuous system were chosen to demonstrate that the technique really was generally applicable except for the case of saturation, when creating inverses.

8.5 Some standard results

In previous chapters a comprehensive range of standard results and their behaviour-patterns were investigated. Not all of them need to be considered here either because some pairs of nonlinearities act as mutual inverses or for reasons which will be explained as each case is considered. Saturation is a non-linearity which relates to the intrinsic ability of a system to respond over all possible ranges of input signal. Now, since all systems have physical

limitations which govern their response ranges no software technique can get around that problem. This feature is discussed later in this chapter in section 8.6.2.1. Dead-zone on its own has not been considered. If a system only consisted of the dead-zone non-linearity then it would never respond to any input. Also, although dead-zone followed by a linear response is common no limit-cycling would occur. However, in the case of a dead-zone followed by a linear response followed by saturation, which is a much more realistic proposition, then a limit-cycle would occur. If the dead-zone can be completely removed, but not the saturation, then the limit-cycle response should change to that of saturation on its own. Since the describing functions for soft and for hard saturation were similar the soft saturation case was chosen as the starting-point of the standard cases presented here.

8.5.1 Dead-zone plus soft saturation

In Figure 7.5 the settings for a dead-zone plus linear section plus hard saturation module were displayed. However, when it comes to designing an inverse, two further practical considerations have to be taken into account (i) as will be explained more fully in section 8.6.3.1, there is a physical limit on the response to hard saturation and so when designing an inverse the case of dead-zone plus linear section plus SOFT saturation has been considered here (ii) there is also a software consideration. In order for the inverse to operate correctly the response of the software must be declared to exist over the complete operating range of the investigation. The settings for the required non-linearity are shown in Figure 8.11. From these a simple mirror image was designed and the results are also shown in Figure 8.11. The non-linearity

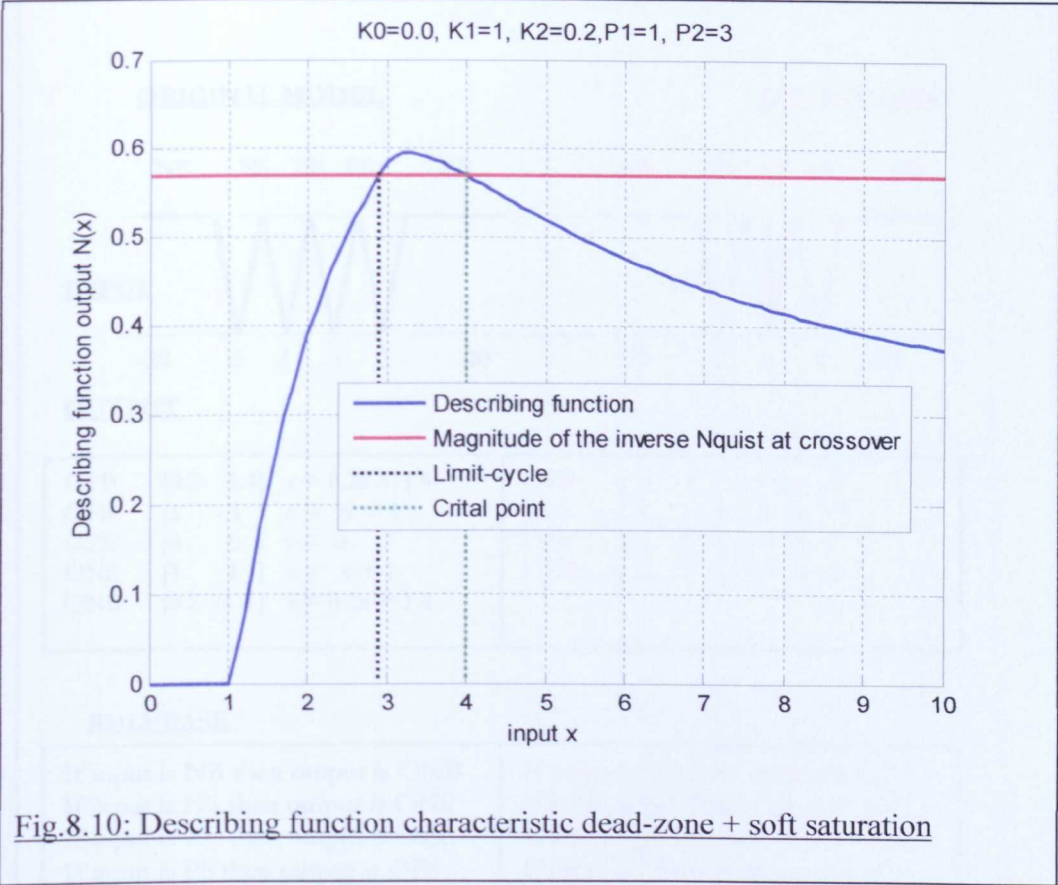
drawn, the mirror image created and then the settings for the fuzzy inverse calculated.

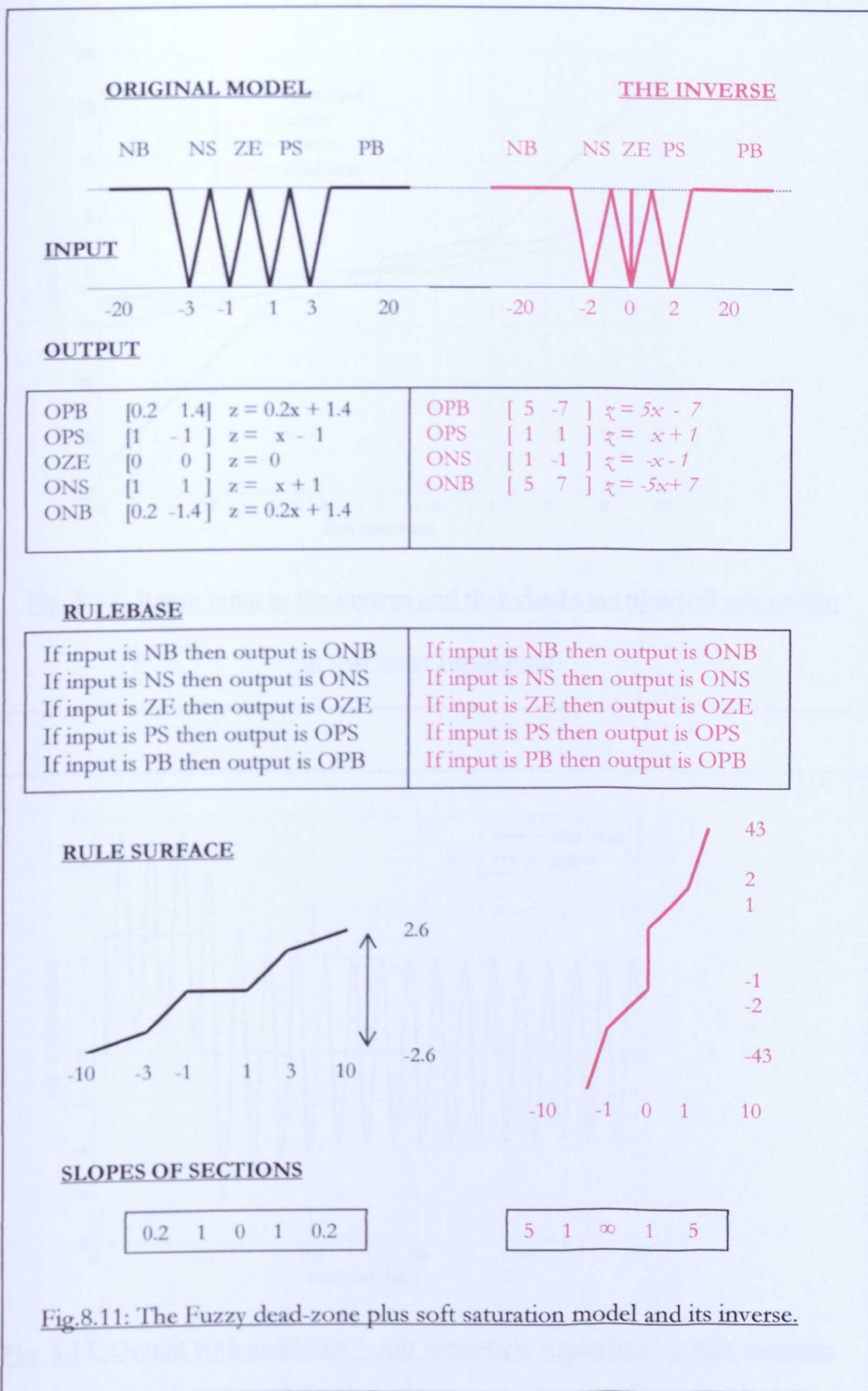
After it had been designed, the fuzzy inverse module was tested against its original non-linear counterpart as described in section 8.4.1. In the original experiment, when only the fuzzy soft saturation module had been used, the output had been a limit cycle, Figure 7.4. However, this time, with the inverse also present, the result was runaway oscillation exactly the same as had been achieved in the original circuit, Figure 5.1, when the transfer function was on its own

A general comment about fuzzy logic simulation is relevant at this stage. Care had to be taken to make sure that the fuzzy signals had been defined over large enough ranges. If they had not been then the resulting output could be in an undefined range and the observed output would usually be zero, or sometimes something completely spurious.

8.5.2 Colomb friction plus viscous drag

As explained earlier, Coulomb friction and dead-zone were mutual inverses. Therefore in finding the inverse for the dead-zone we automatically had the inverse of the Coulomb friction case – without further manipulation.





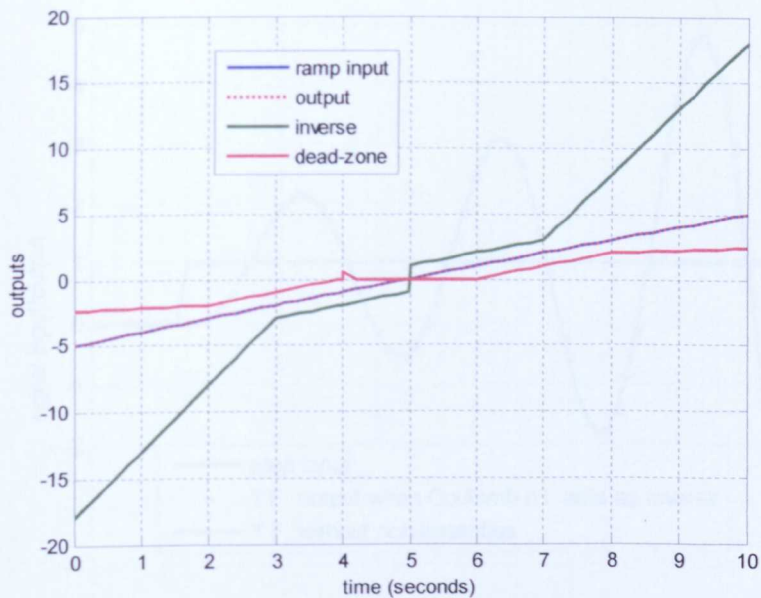


Fig. 8.12: Ramp input to the inverse and then deadzone plus soft saturation, in that order (open loop)

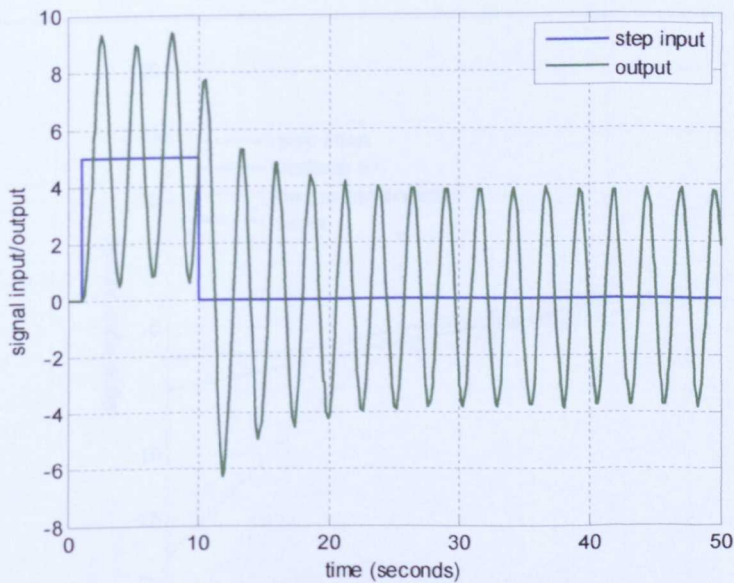


Fig. 8.13: Output with deadzone + soft saturation + standard transfer function

Note: In Figure 8.14 care had to be taken to ensure the fuzzy modules had been defined with a wide-enough operating range (± 40 in this instance).

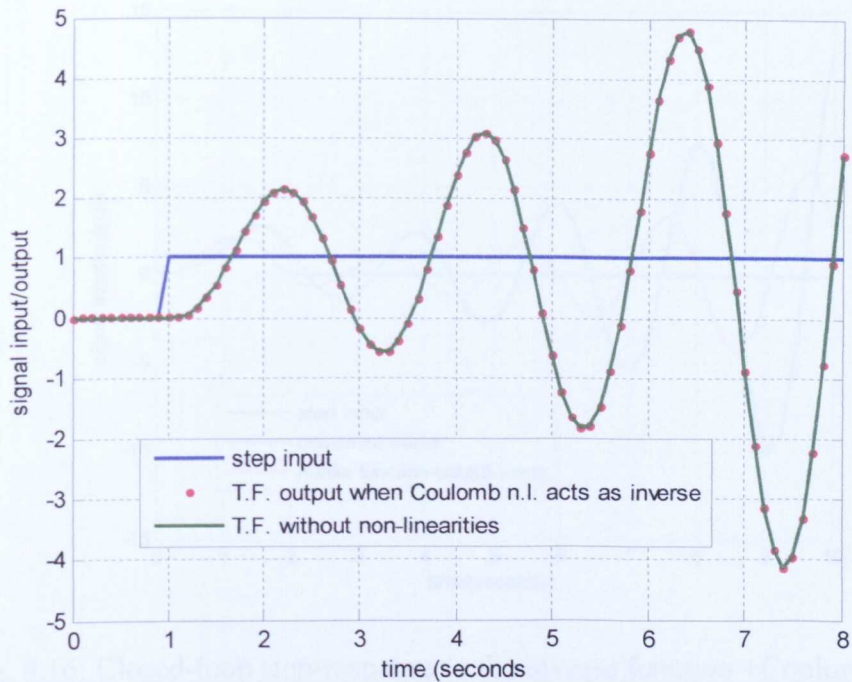


Fig. 8.14: Step response to inverse + dead-zone in series with standard transfer function

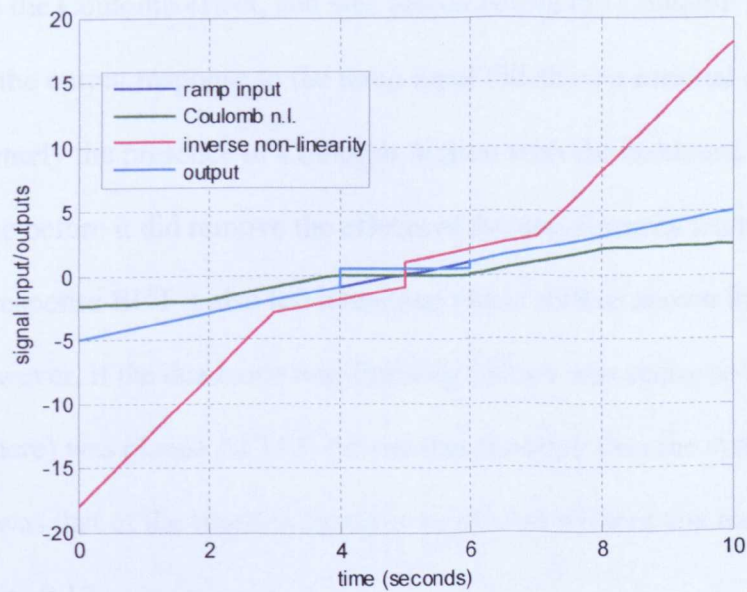


Fig.: 8.15: Ramp input to Coulomb friction and its inverse, (open loop). Inverse between ramp input and the Coulomb friction

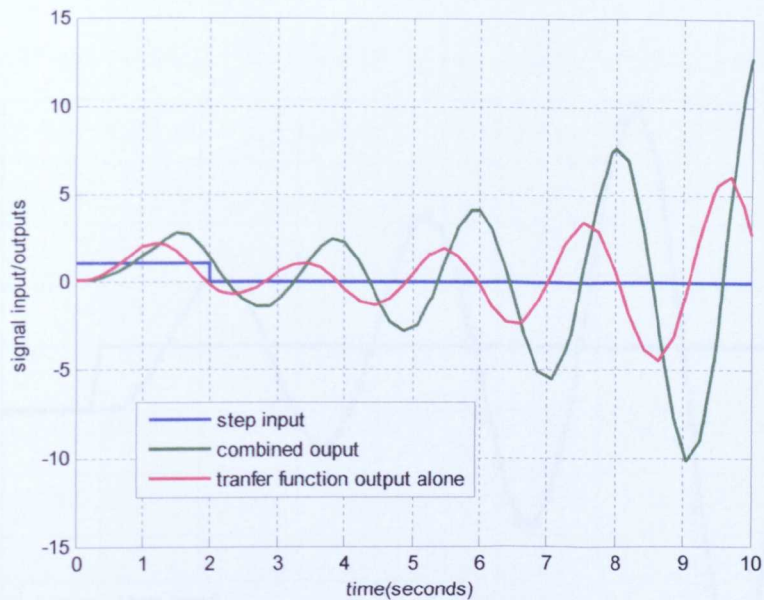


Fig. 8.16: Closed-loop step-response to the inverse function +Coulomb non-linearity in series with the standard transfer function

From Figure 8.15, when the deadzone non-linearity was acting as the inverse to the Coulomb effect, and was placed before the Coulomb non-linearity, the output response to the ramp input did show a residual deadzone. Consequently the presence of Coulomb friction with the deadzone acting as the inverse before it did remove the effects of the non-linearity from the transfer function response BUT it also left a residual phase shift as shown in Figure 8.16. However, if the deadzone non-linearity (which was acting as the inverse function here) was placed AFTER the transfer function then the system response was that of the transfer function on its own without any residual phase shift, Figure 8.17.

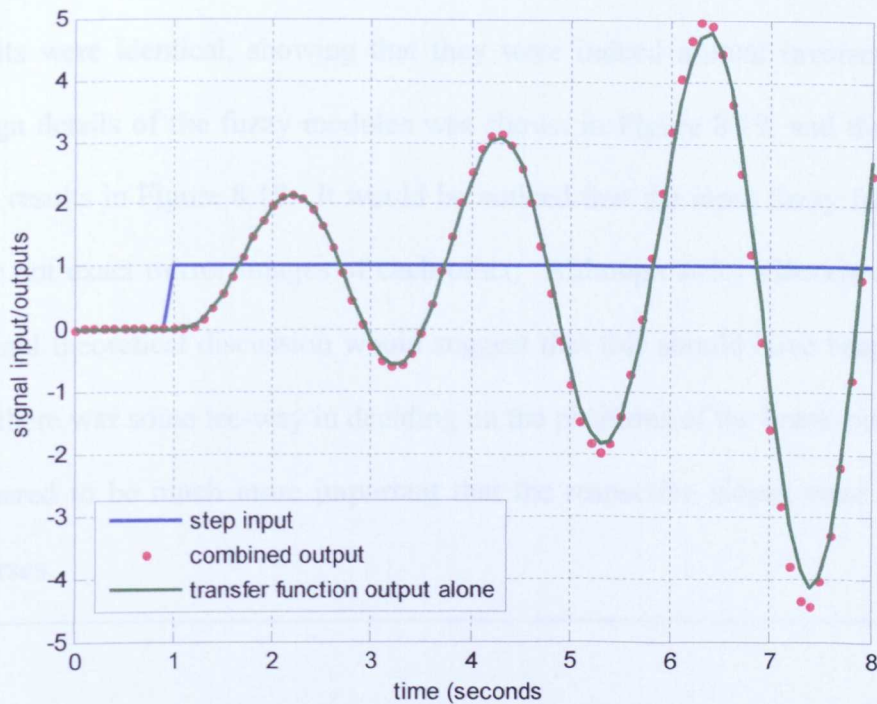


Fig. 8.17: Coulomb + inverse in series with standard transfer function
with the inverse function placed after the transfer function

8.5.3 The triple-slope non-linearity

Only case (ii), $K_0 > K_1$ & $K_2 > K_1$, was shown as cases (ii) and (iii) would be mutual inverses and case (i) added nothing extra to the demonstration of the method. After the models had been designed, the fuzzy inverse module, case (ii), was then placed in series with its inverse and a unit ramp input applied to this open-loop arrangement. The output was a unit ramp identical to the input signal, Figure 8.18. The two fuzzy modules were then placed in a closed loop circuit as in Figure 8.9 and a step-input applied. In the original experiment, when only the case (ii) module had been used, the output had been a limit cycle, Figure 7.15. However, this time the result was runaway oscillation exactly the

same as had been achieved in the original circuit, Figure 5.1, when no non-linearity had been present. The roles of the two modules were reversed and the results were identical, showing that they were indeed mutual inverses. The design details of the fuzzy modules was shown in Figure 8.19. and the open-loop results in Figure 8.18. It would be noticed that the input fuzzy functions were not exact mirror images of each other. Although strict adherence to the original theoretical discussion would suggest that this should have been so, in fact there was some lee-way in deciding on the positions of the break-points. It appeared to be much more important that the respective slopes were mutual inverses

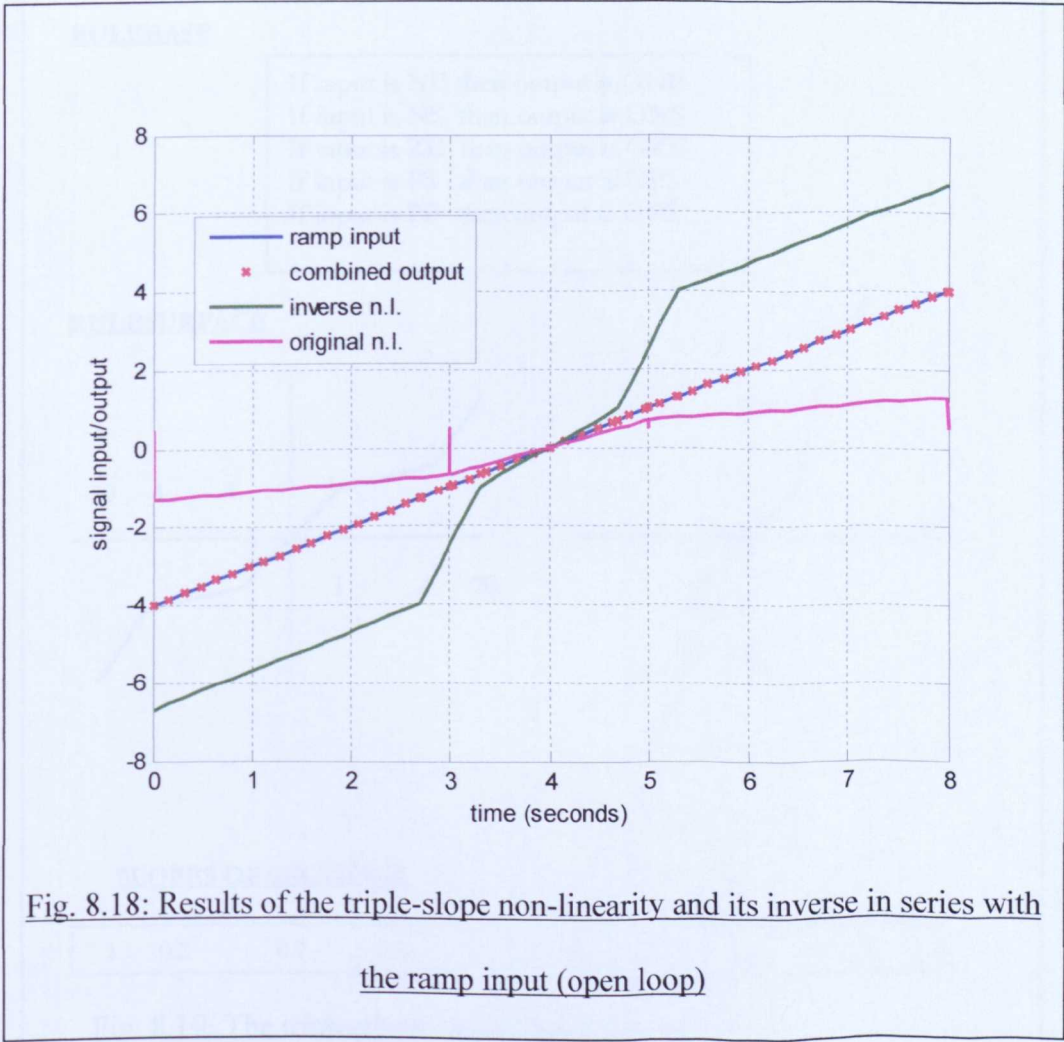


Fig. 8.18: Results of the triple-slope non-linearity and its inverse in series with the ramp input (open loop)

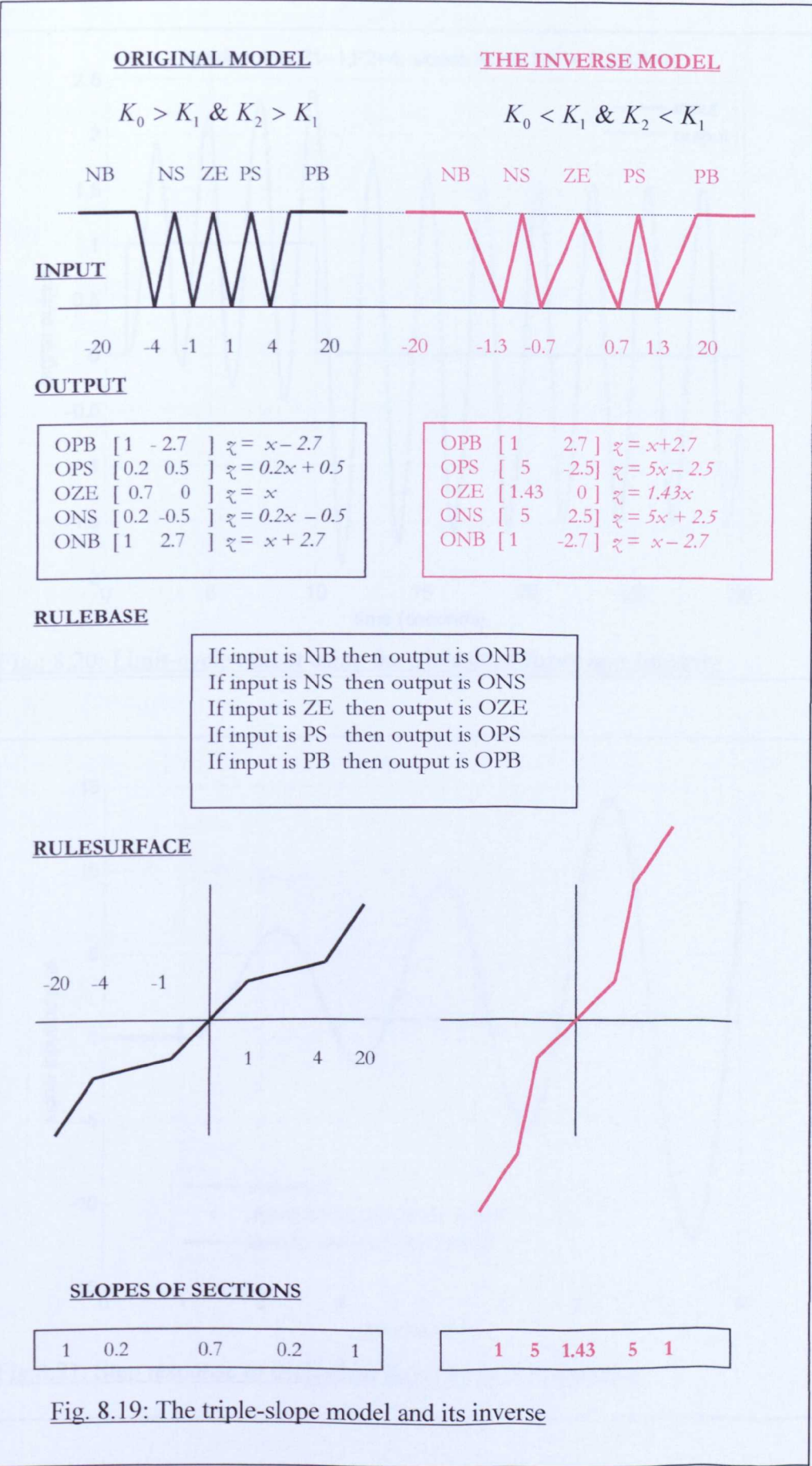
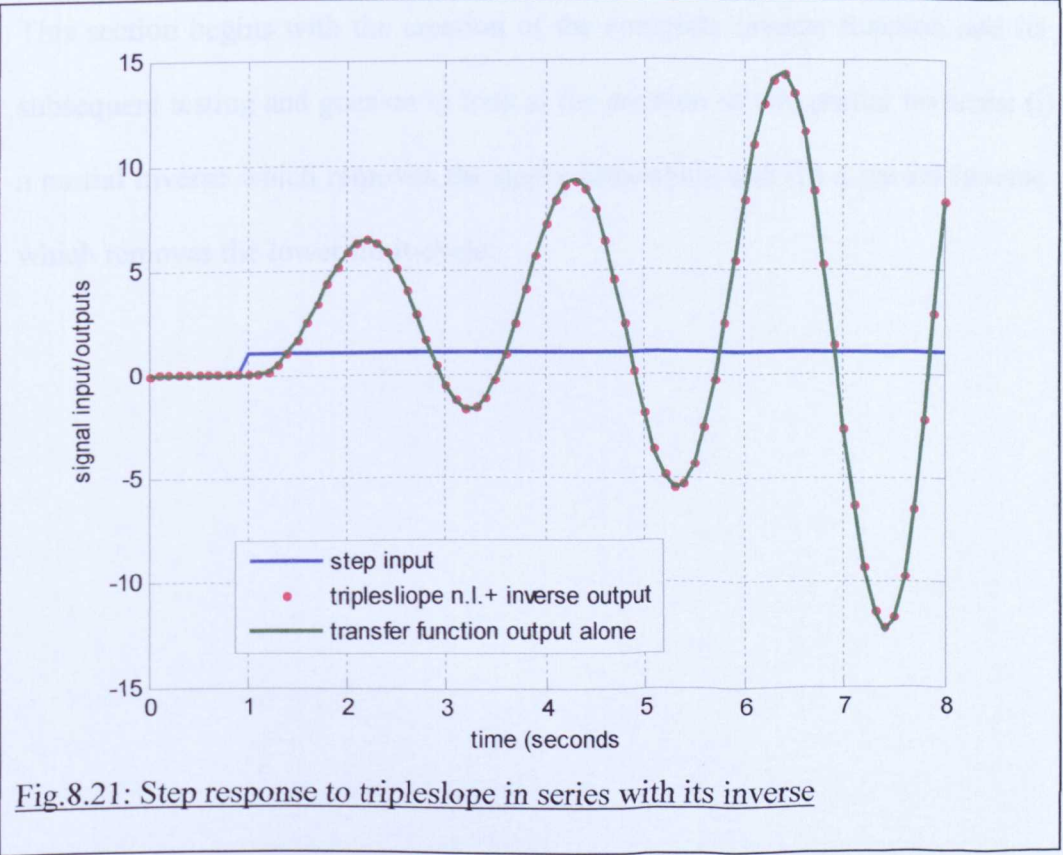
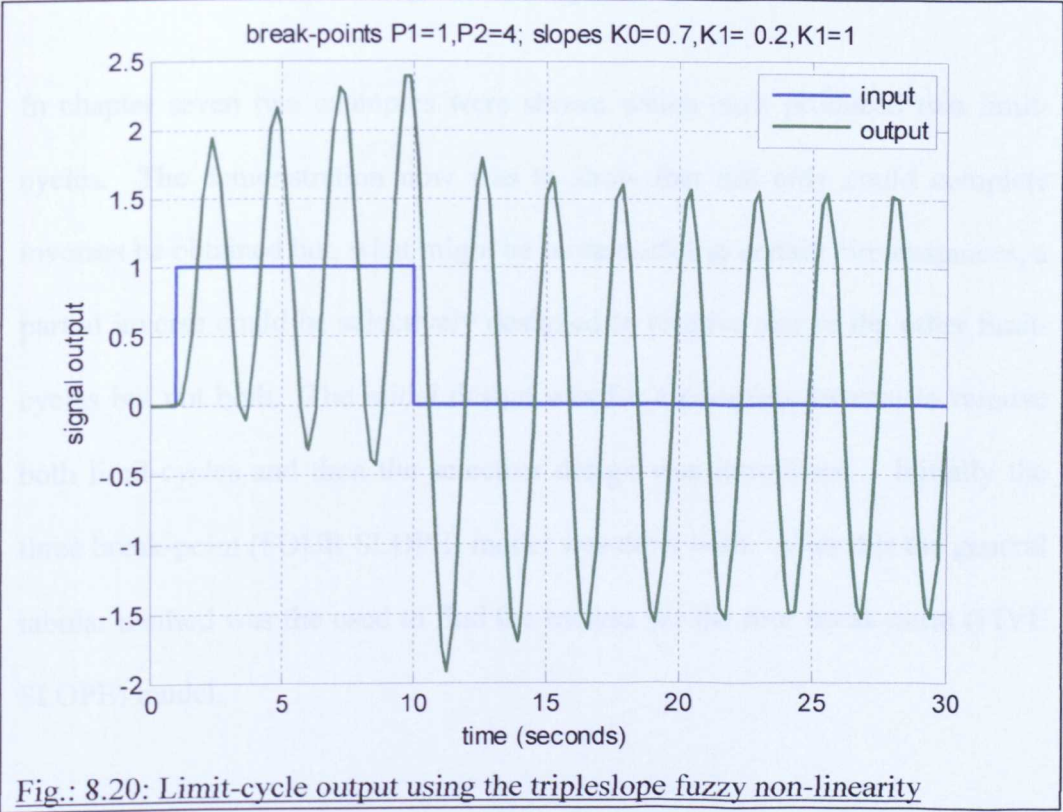


Fig. 8.19: The triple-slope model and its inverse



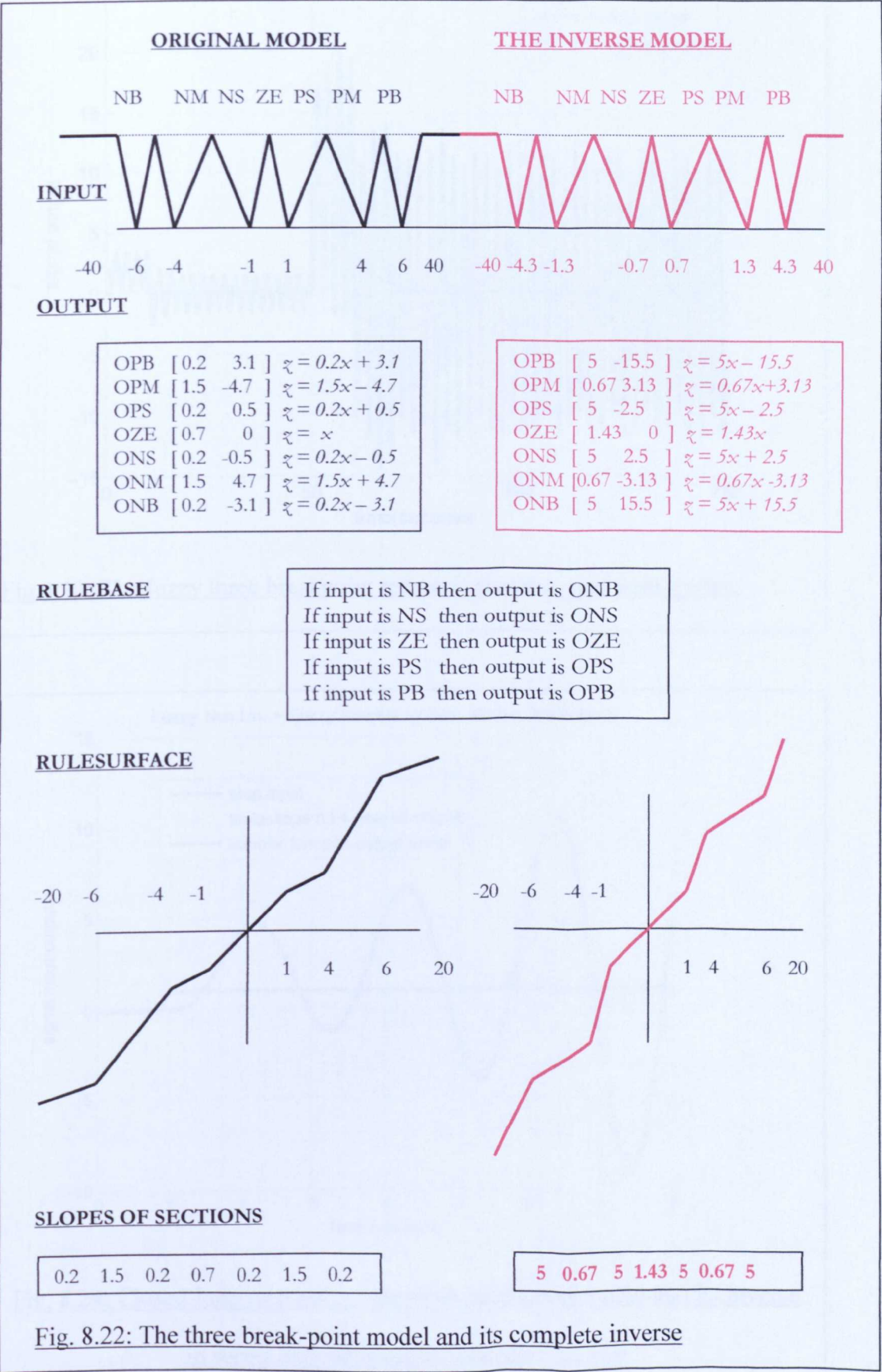
8.6 The double limit-cycle examples

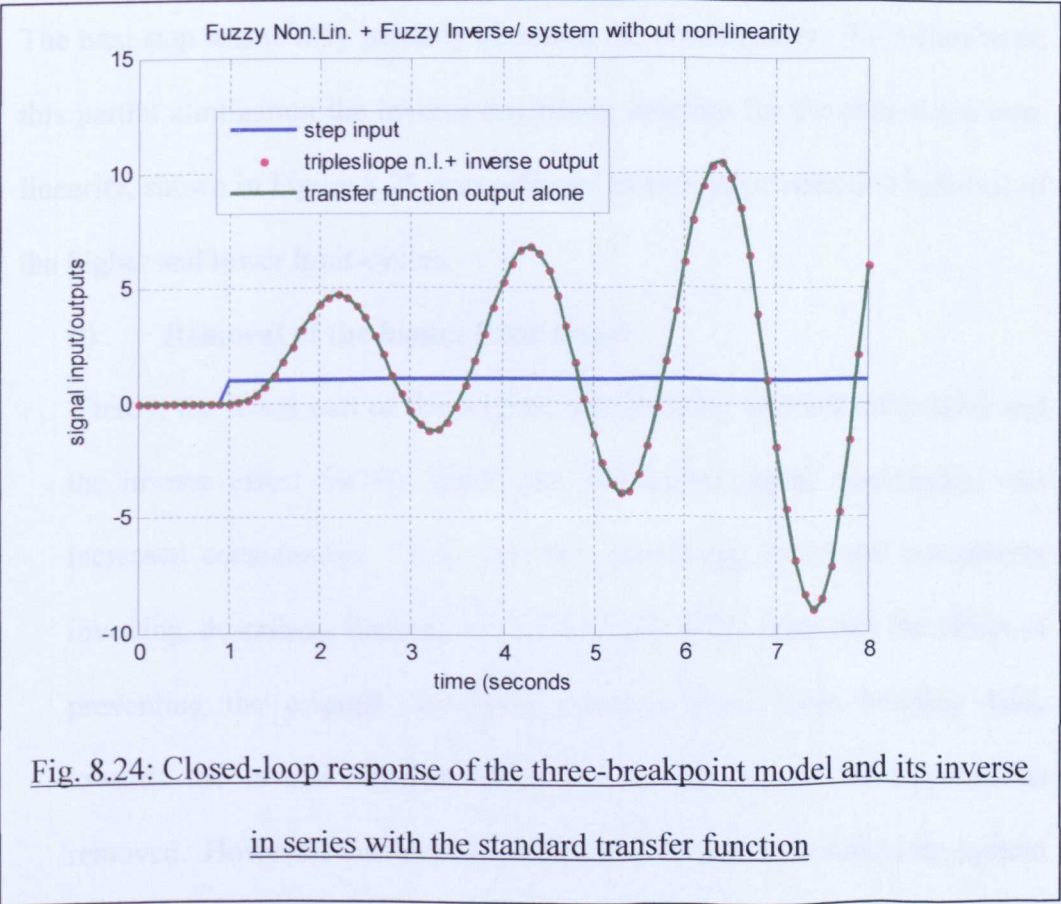
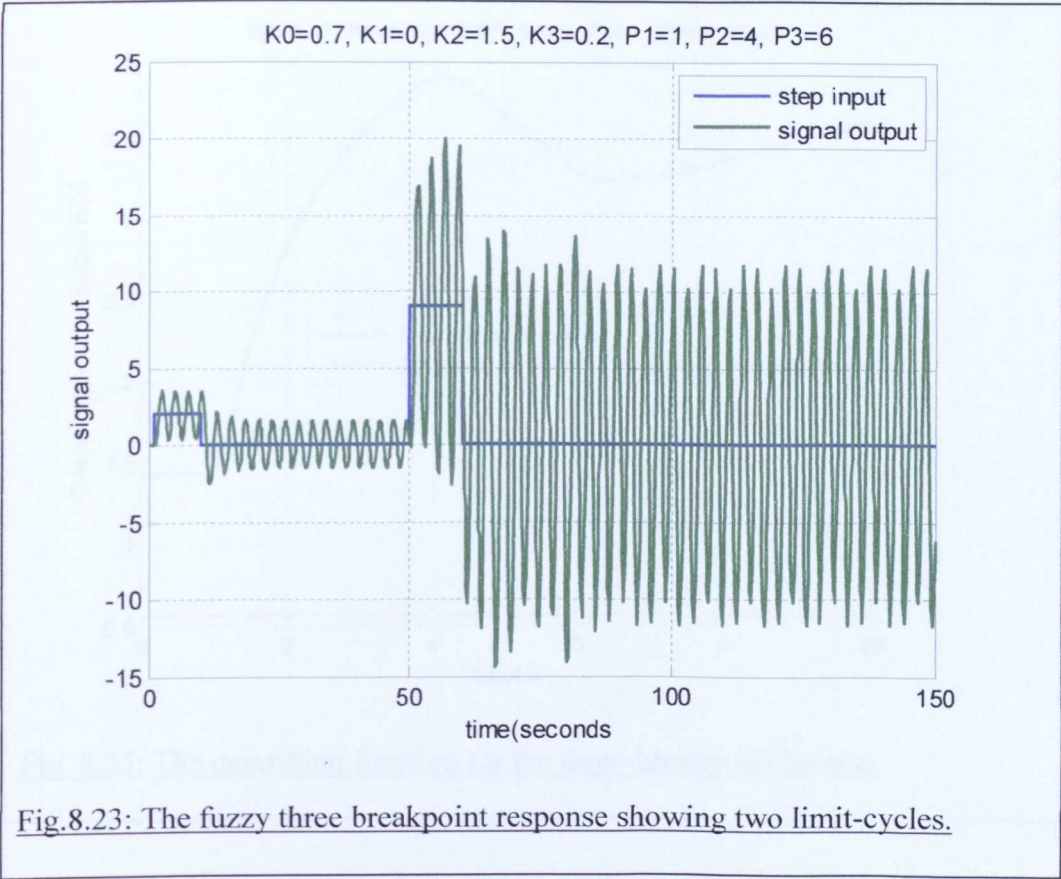
In chapter seven two examples were shown which each produced two limit-cycles. The demonstration now was to show that not only could complete inverses be obtained but, what might be more useful in certain circumstances, a partial inverse could be selectively designed to remove one or the other limit-cycles but not both. The initial design was for a complete inverse to remove both limit-cycles and then the selective design was introduced.. Initially the three break-point (FOUR SLOPE) model was dealt with. After this the general tabular method was the used to find the inverse for the four break-point (FIVE SLOPE) model.

8.6.1 The three break-point (four-slope) non-linearity

This section begins with the creation of the complete inverse function and its subsequent testing and goes on to look at the creation of two partial inverses: (i) a partial inverse which removes the upper limit-cycle and (ii) a partial inverse which removes the lower limit-cycle.

8.6.1.1 The complete inverse function





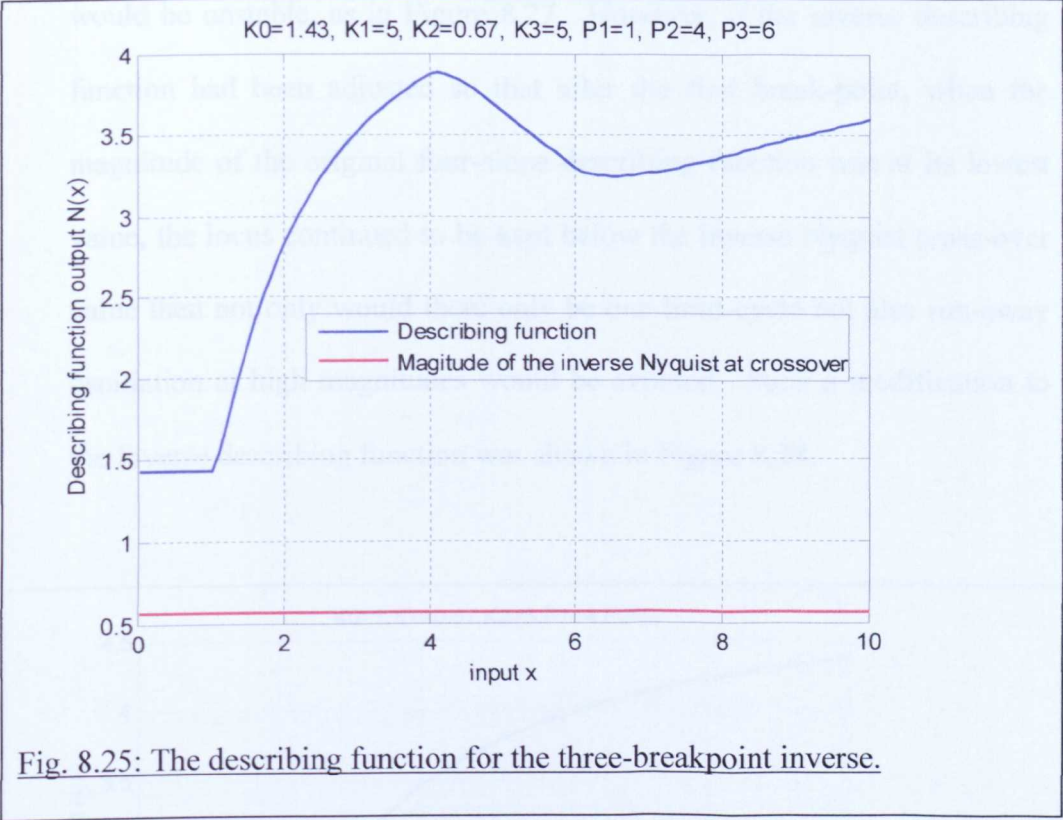


Fig. 8.25: The describing function for the three-breakpoint inverse.

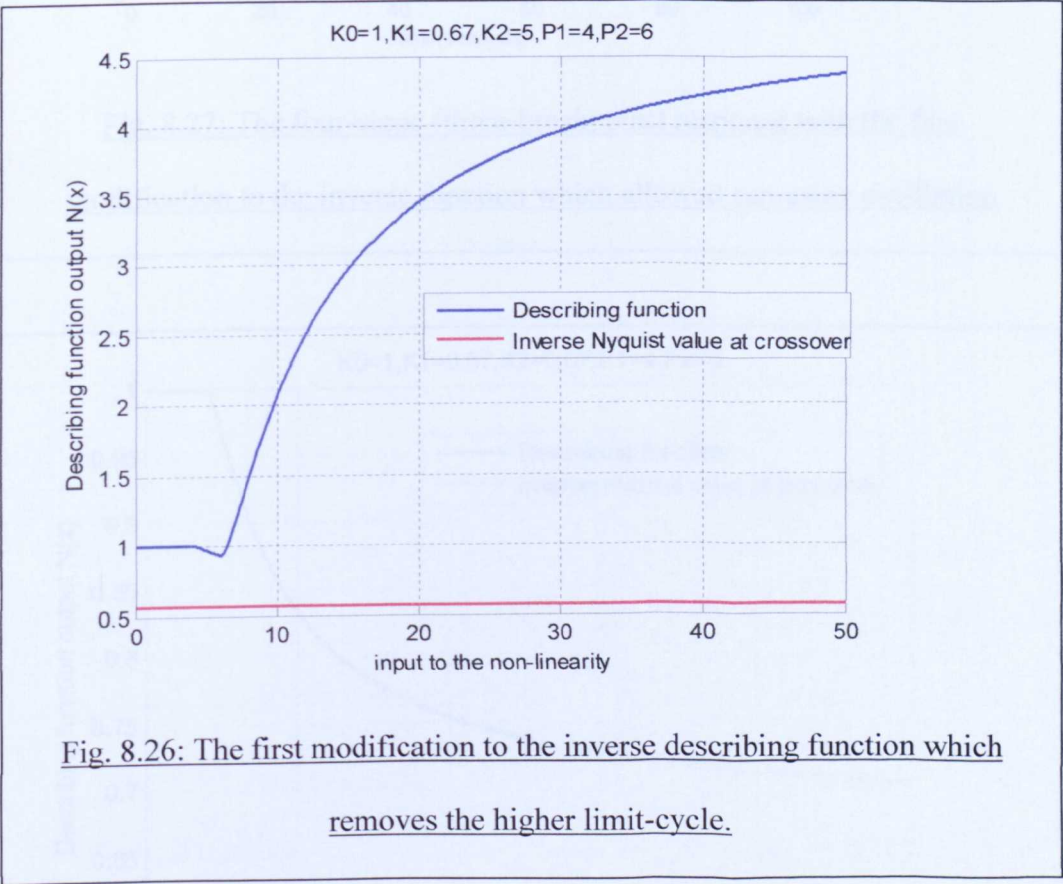
8.6.1.2 Partial inverse functions

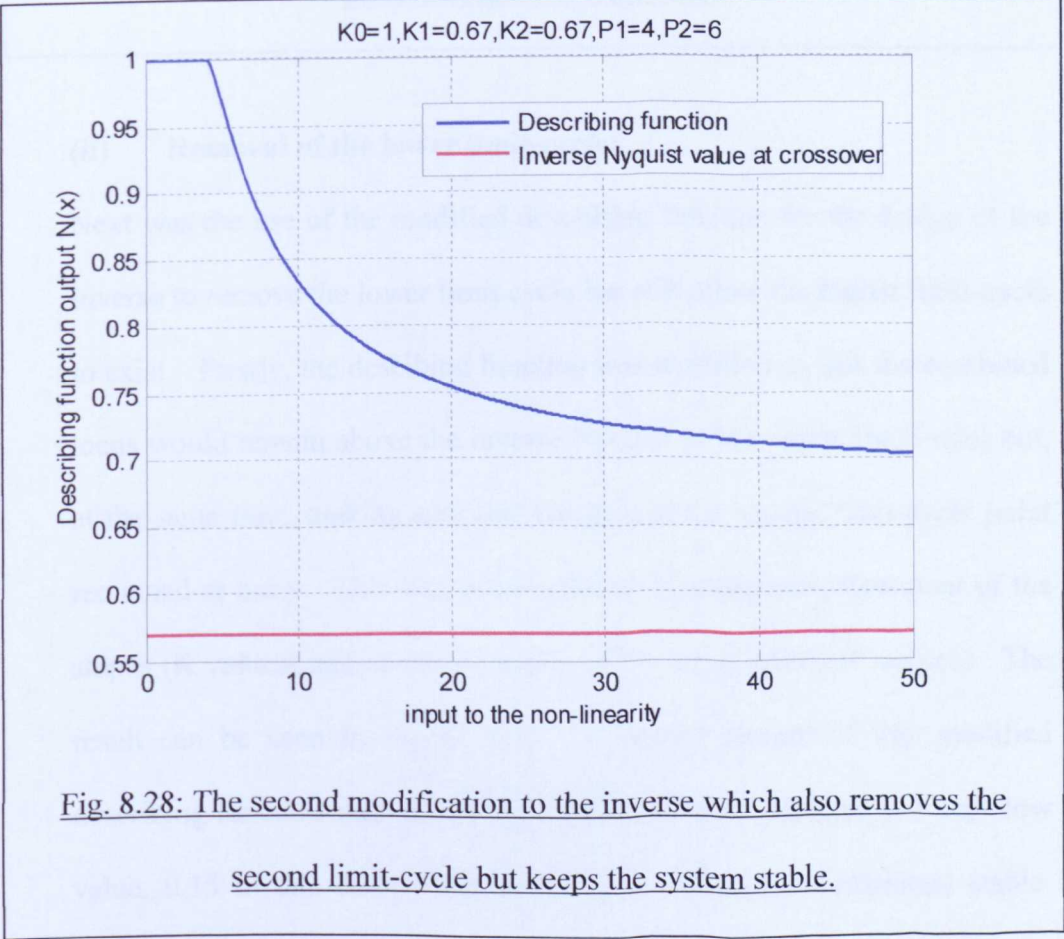
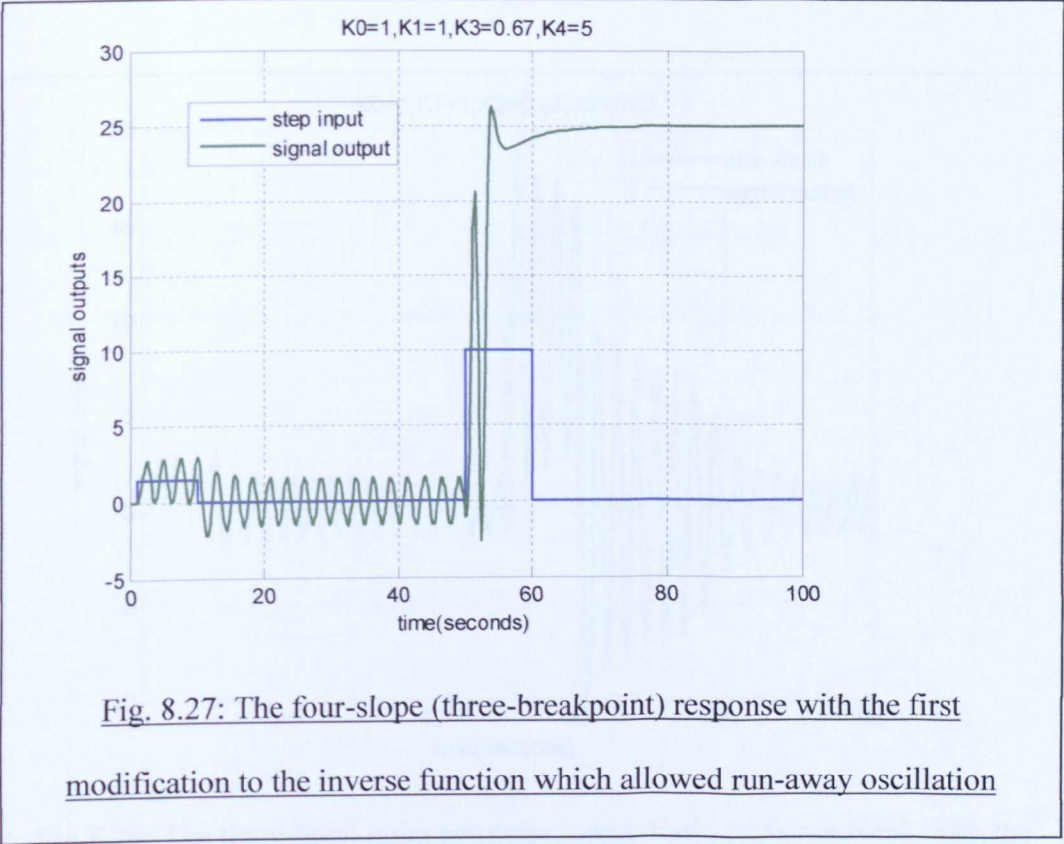
The next step was to only partially eliminate the non-linearity. To demonstrate this partial elimination the inverse describing function for the four-slope non-linearity, shown in Figure 8.25, was adjusted in two ways: selective removal of the higher and lower limit-cycles.

(i) Removal of the higher limit-cycle:

Firstly, the lower part of the original non-linearity was left untouched and the inverse effect for the upper part, the higher signal amplitudes, was increased considerably. This gave the modifying, no longer completely inverting, describing function shown in Figure 8.26. This had the effect of preventing the original describing function locus from bending back towards the inverse Nyquist value and so the second limit-cycle was removed. However, this had the effect that for high input values the system

would be unstable, as in Figure 8.27. However, if the inverse describing function had been adjusted so that after the first break-point, when the magnitude of the original four-slope describing function was at its lowest value, the locus continued to be kept below the inverse Nyquist cross-over value then not only would there only be one limit-cycle but also run-away oscillation at high magnitudes would be avoided. Such a modification to the inverse describing function was shown in Figure 8.28.





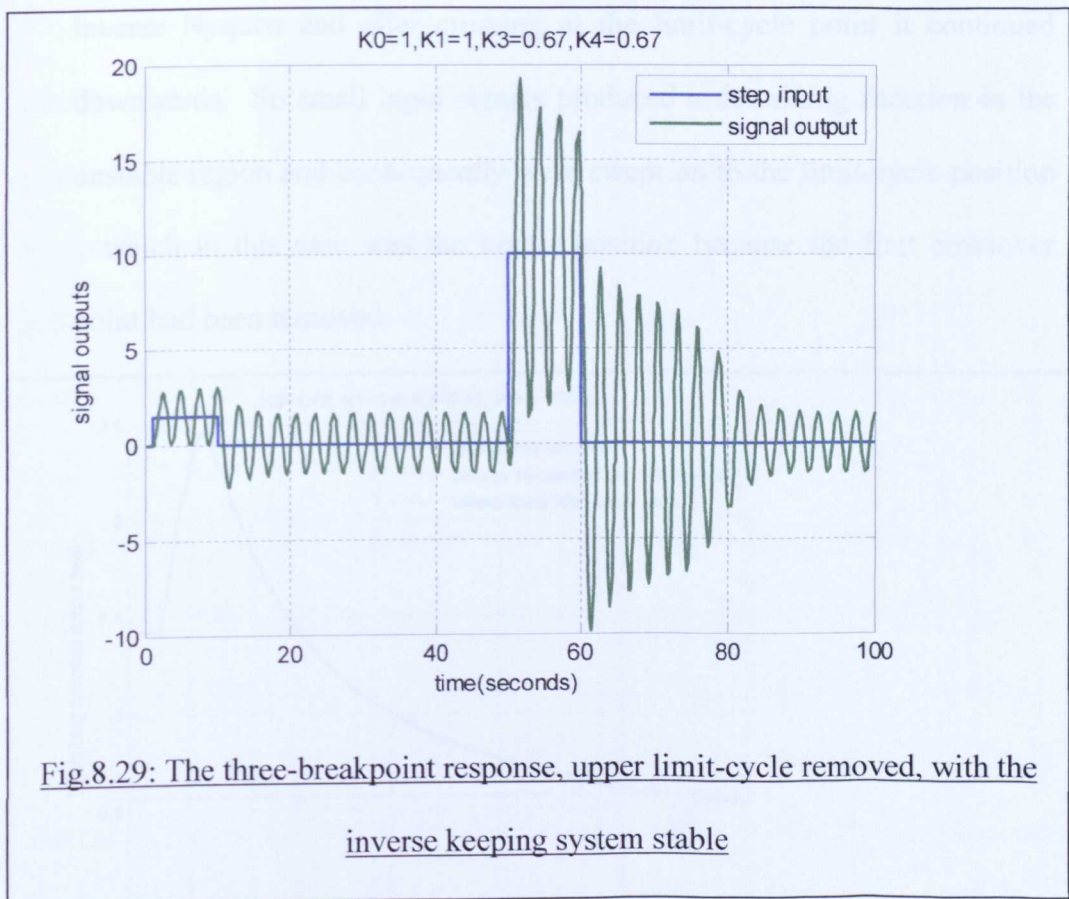


Fig.8.29: The three-breakpoint response, upper limit-cycle removed, with the inverse keeping system stable

(ii) Removal of the lower limit-cycle:

Next was the use of the modified describing function for the design of the inverse to remove the lower limit cycle but still allow the higher limit-cycle to exist. Firstly, the describing function was modified so that the combined locus would remain above the inverse Nyquist at low input amplitudes but, at the same time, making sure that the gain at the second limit-cycle point remained at unity. This was accomplished by judicious adjustment of the slopes (K values) and of the positions of the break-points (P values). The result can be seen in Figure 8.30. A further feature of this modified describing function was that at high signal values it decayed to a very low value, 0.15 in this case. This ensured that the system remained stable.

However, the describing function started at a value greater than that of the inverse Nyquist and after crossing at the limit-cycle point it continued downwards. So small input signals produced a describing function in the unstable region and consequently were swept on to the limit-cycle position – which in this case was the higher position because the first crossover point had been removed.

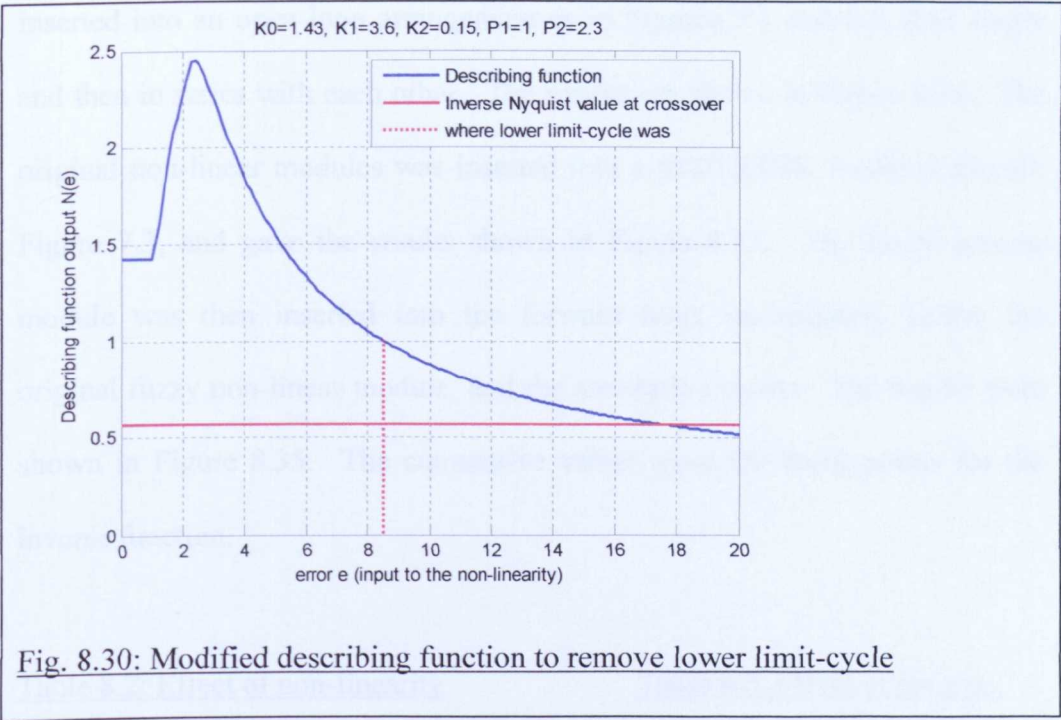


Fig. 8.30: Modified describing function to remove lower limit-cycle

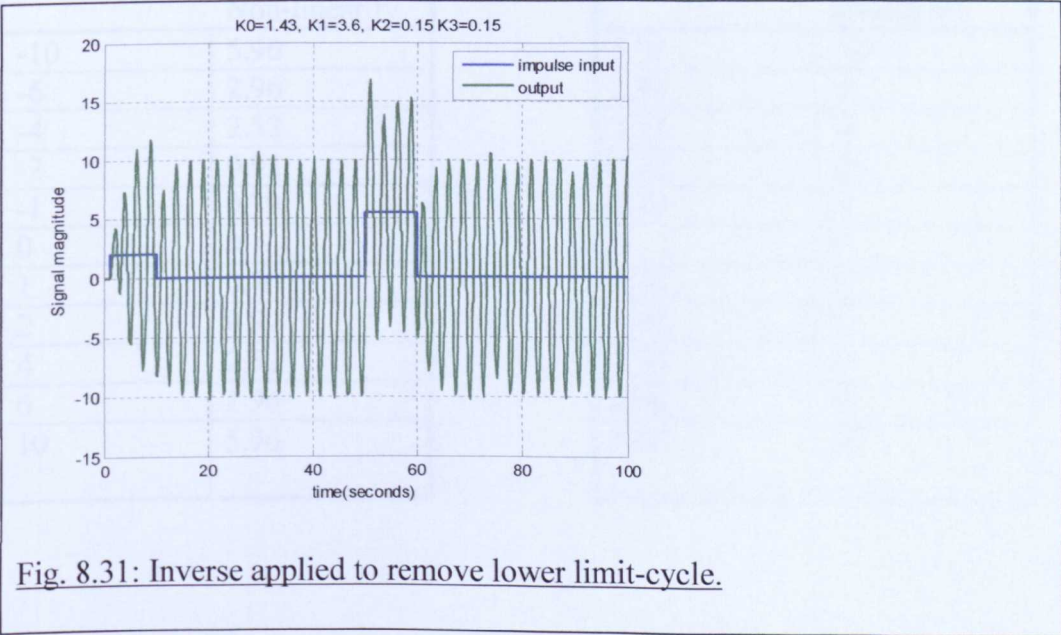


Fig. 8.31: Inverse applied to remove lower limit-cycle.

8.6.2 The four breakpoint case

An inverse was created for the four-breakpoint (five slope) example from section 5.3.2. Table 8.2 gave the input values and the cumulative amplifications introduced by the non-linearity, whilst Table 8.3 gave the same story but with the axes effectively reversed. The fuzzy non-linear modules relating to these tables were then created, Figure 8.32, and these modules inserted into an open-loop arrangement as in Figures 7.1 and 8.6, first singly and then in series with each other. The results are shown in Figure 8.34. The original non-linear modules was inserted into a SIMULINK feedback circuit, Figure 7.3, and gave the results shown in Figure 8.33. The fuzzy inverse module was then inserted into the forward loop, immediately before the original fuzzy non-linear module, and the simulation re-run. The results were shown in Figure 8.35. The cumulative values gave the break-points for the inverse function.

Table 8.2: Effect of non-linearity

Input signal	Cumulative Non-linearity
-10	5.96
-6	2.96
-4	2.52
-2	0.92
-1	0.70
0	0
1	0.70
2	0.92
4	2.52
6	2.96
10	5.96

Table 8.3: Effect of inverse

Input signal	Cumulative inverse n-l.
5.96	-10
2.96	-6
2.52	-4
0.92	-2
0.70	-1
0	0
0.70	1
0.92	2
2.52	4
2.96	6
5.96	10

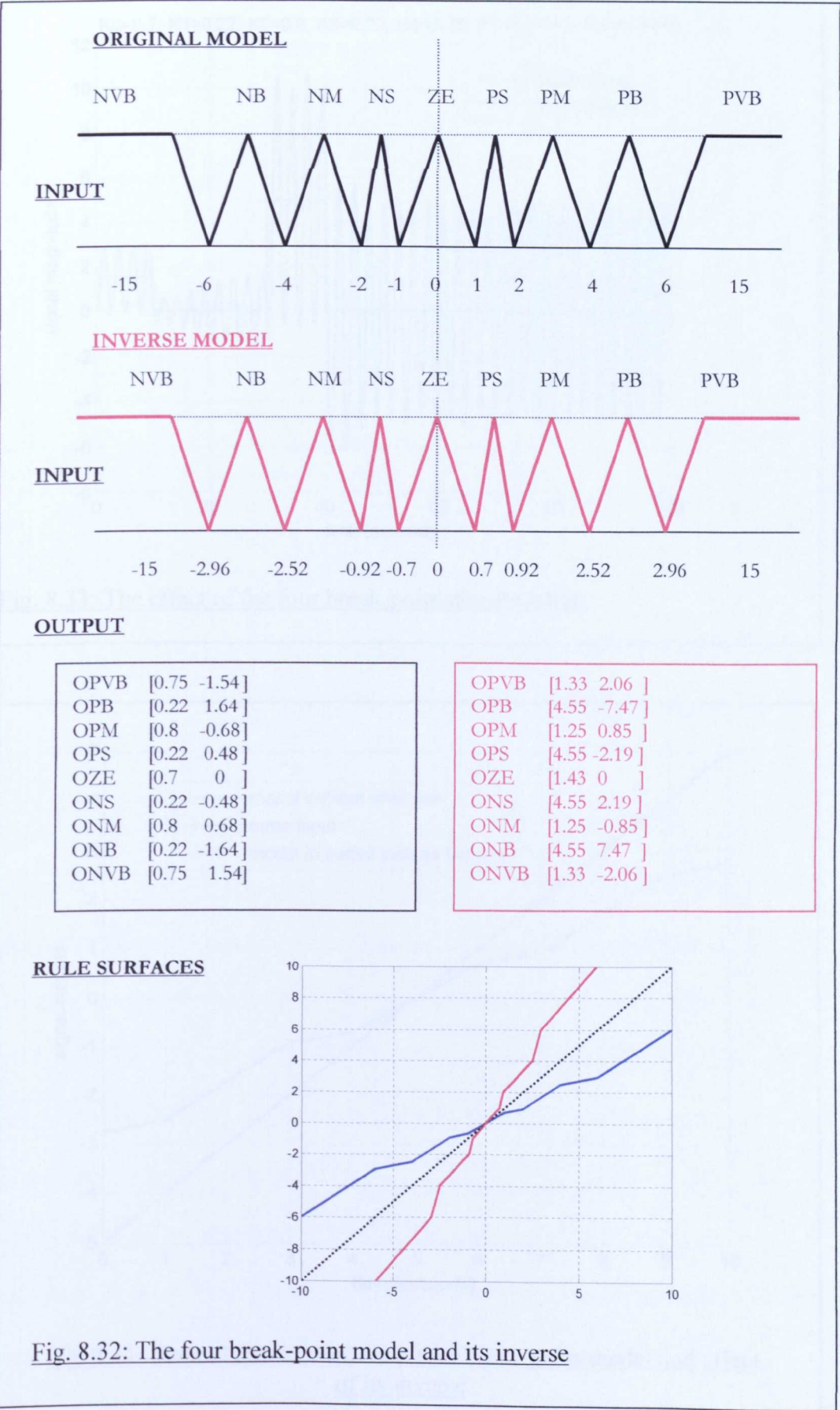
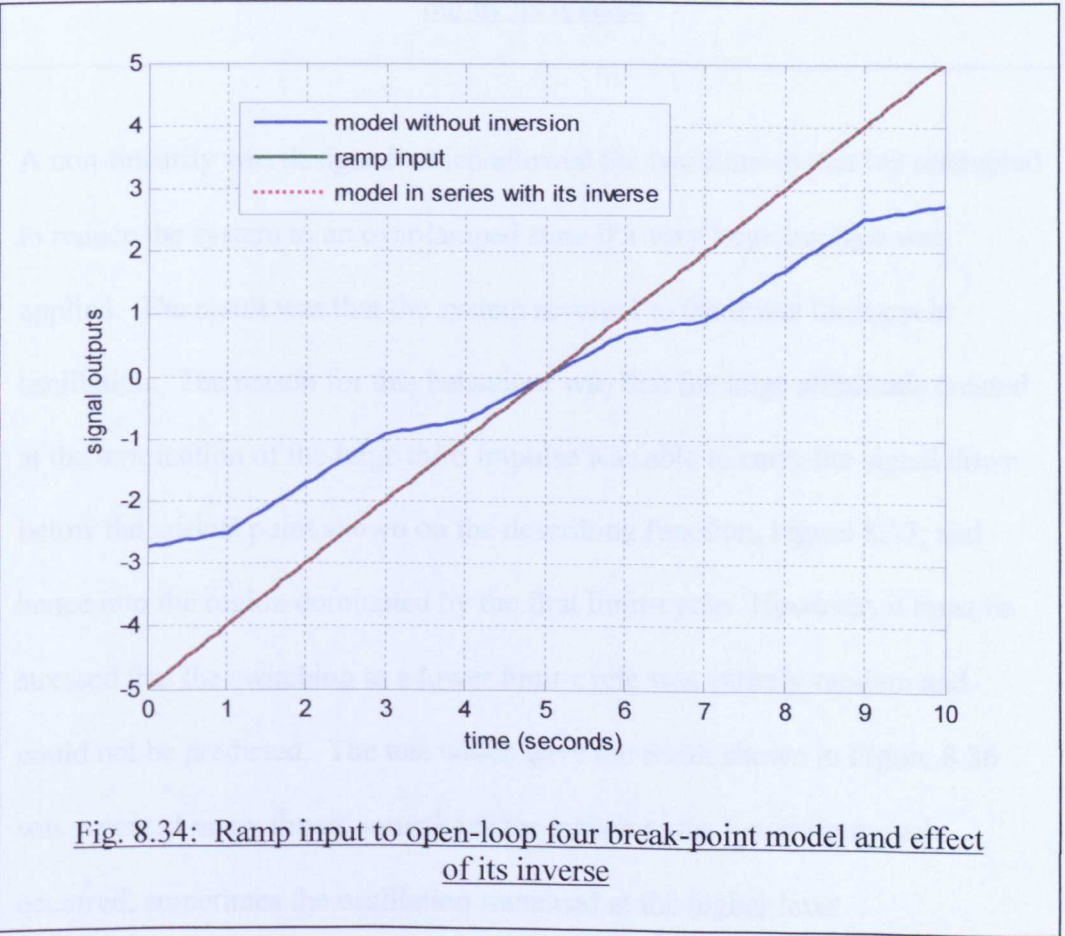
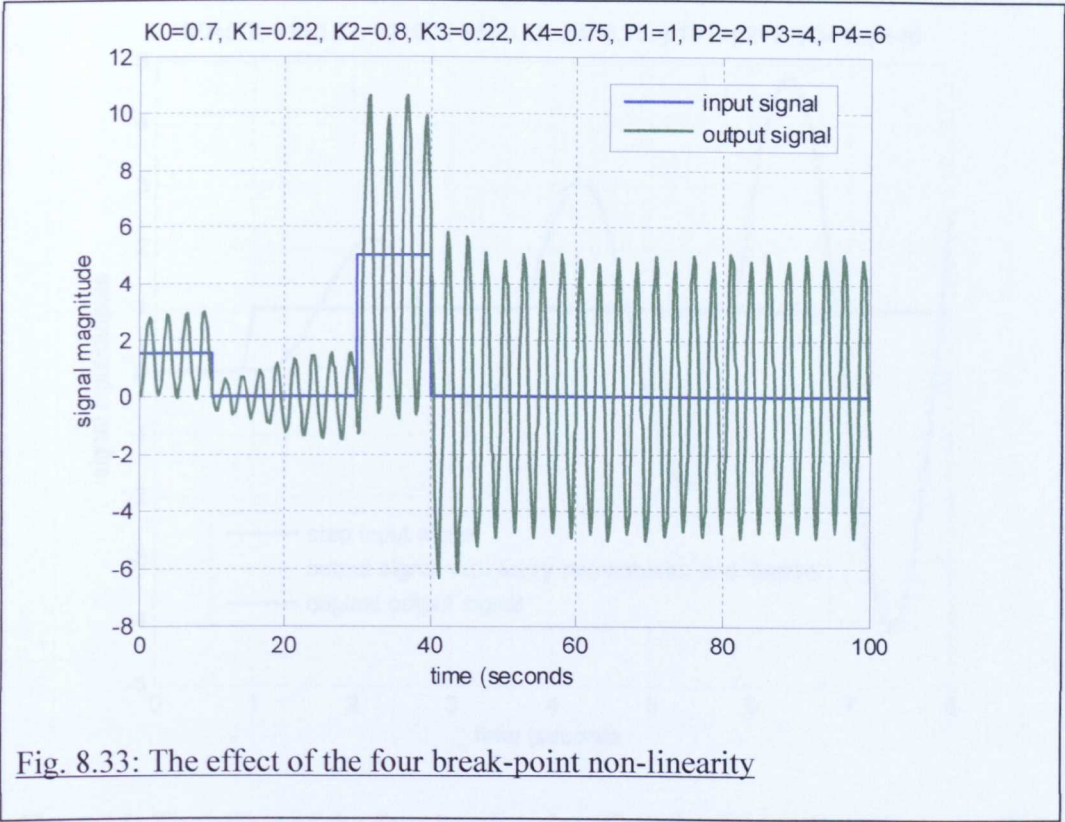
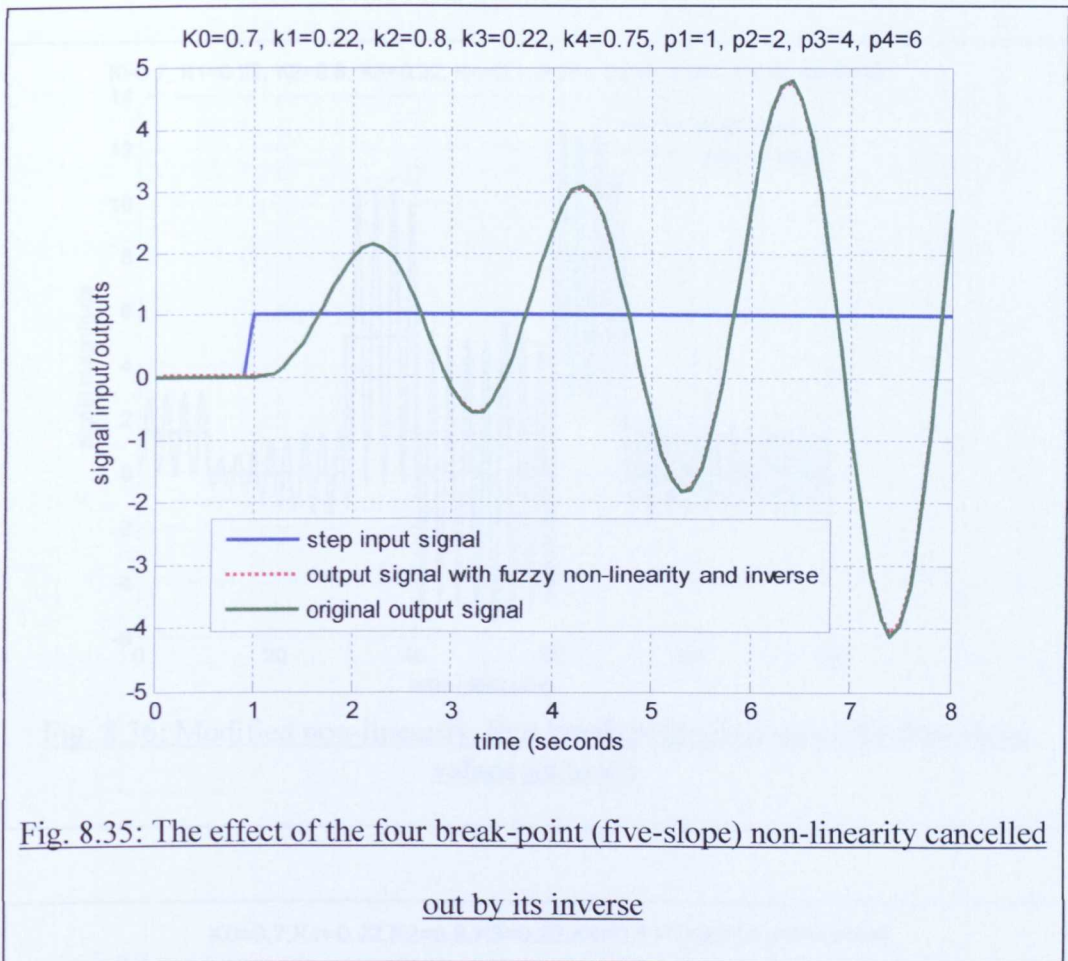
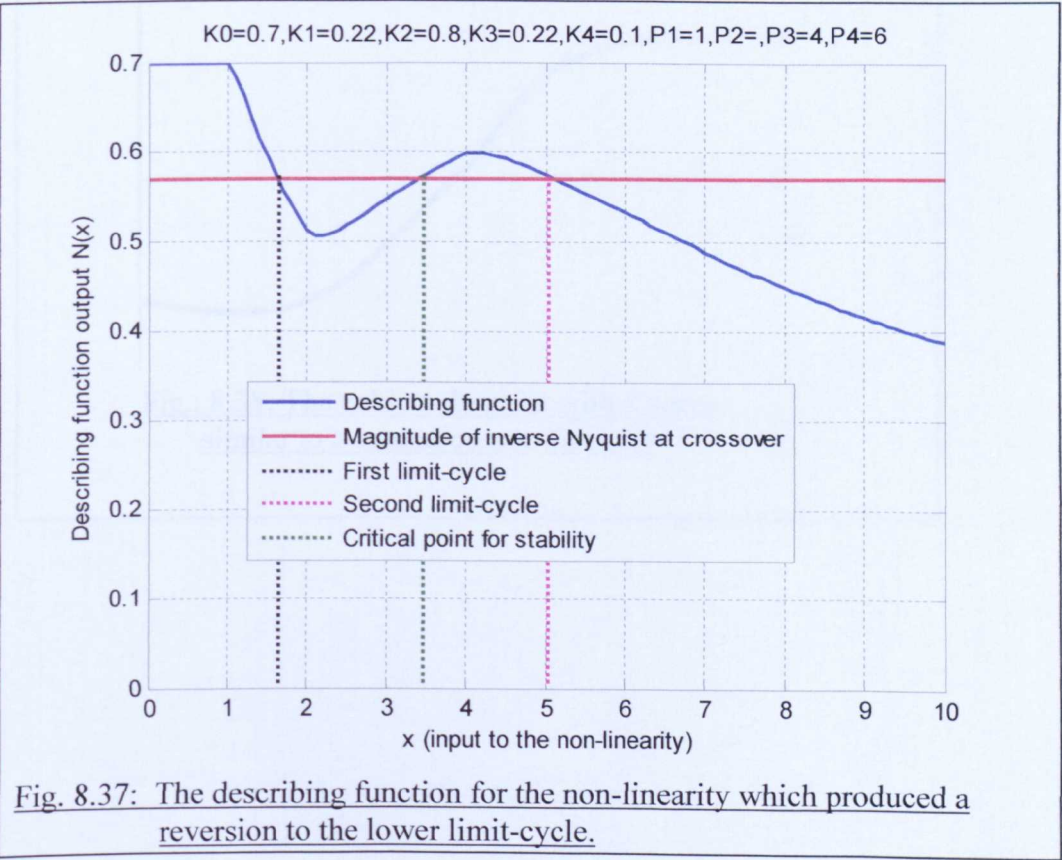
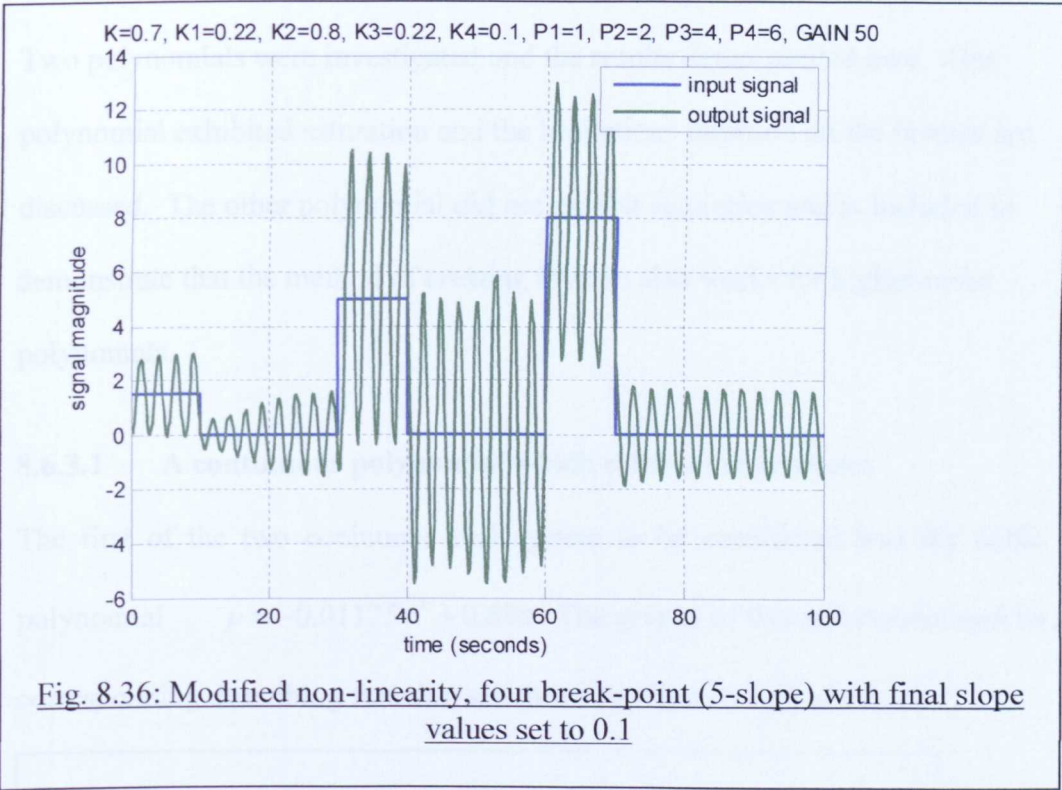


Fig. 8.32: The four break-point model and its inverse





A non-linearity was designed which allowed the two limit-cycles but attempted to reduce the system to an overdamped state if a very large impulse was applied. The result was that the system reverted to the lower limit-cycle oscillation. The reason for this behaviour was that the large amplitude created at the termination of the large third impulse was able to carry the signal down below the critical point shown on the describing function, Figure 8.37, and hence into the region dominated by the first limit-cycle. However, it must be stressed that the switching to a lower limit-cycle was entirely random and could not be predicted. The test which gave the result shown in Figure 8.36 was repeated many times; sometimes the switch to the lower limit-cycle occurred, sometimes the oscillation remained at the higher level.

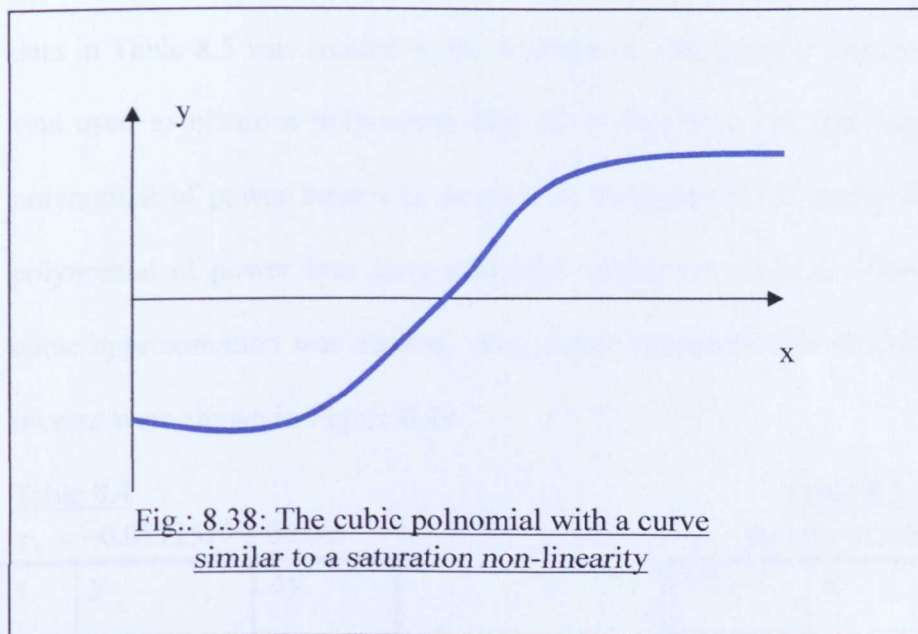


8.6.3 Continuous real non-linear systems

Two polynomials were investigated and the results demonstrated here. One polynomial exhibited saturation and the limitations imposed on the inverse are discussed. The other polynomial did not exhibit saturation and is included to demonstrate that the method of creating inverse also works for higher-order polynomials.

8.6.3.1 A continuous polynomial which exhibits saturation:

The first of the two continuous real systems to be considered was the cubic polynomial $p = -0.01125x^3 + 0.69x$. The graphs of this polynomial and its corresponding describing function are shown in Figures 8.38 and 8.39.



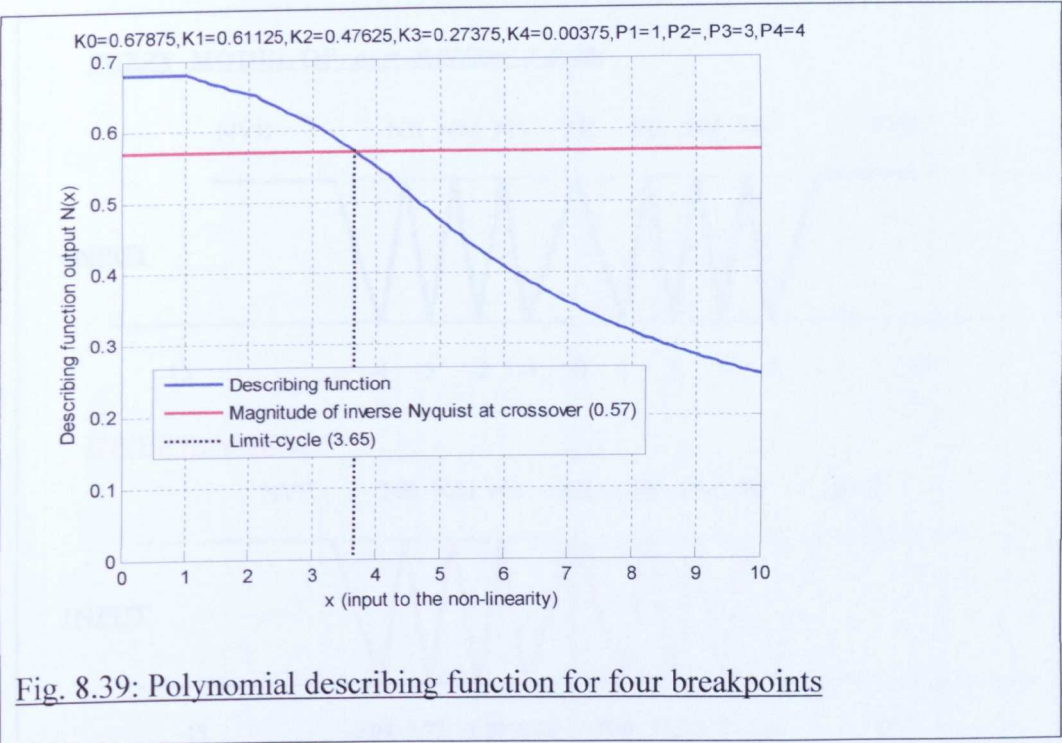


Fig. 8.39: Polynomial describing function for four breakpoints

These results are approximations to the data shown in Table 8.4. From this, the data in Table 8.5 was created as the first step in creating the inverse. Matlab was used to obtain a polynomial best fit to this data. It was found that a polynomial of power three was as good as a polynomial of power five and a polynomial of power four gave excessive values for $x = \pm 4$. Therefore the cubic approximation was chosen. The characteristics of this function and its inverse were shown in Figure 8.40.

Table 8.4

$$p_1 = -0.01125x^3 + 0.69x$$

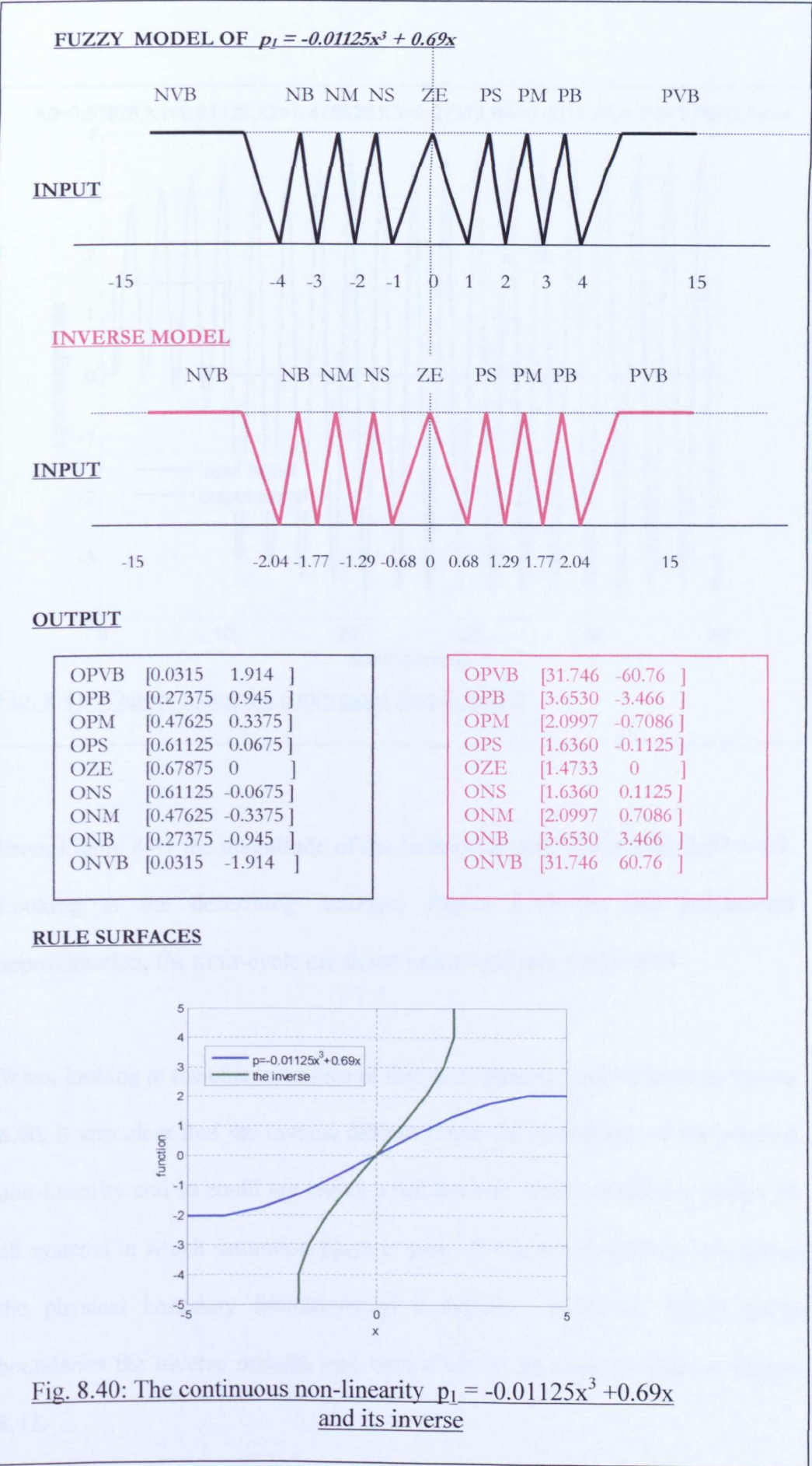
x	y	Δy
5	2.0715	
4	2.04	0.0315
3	1.76625	0.27375
2	1.29	0.47625
1	0.67875	0.61125
0	0.0000	0.67875
-1	-0.67875	0.61125
-2	-1.29	0.47625
-3	-1.76625	0.27375
-4	-2.04	0.0315
-5	-2.04375	

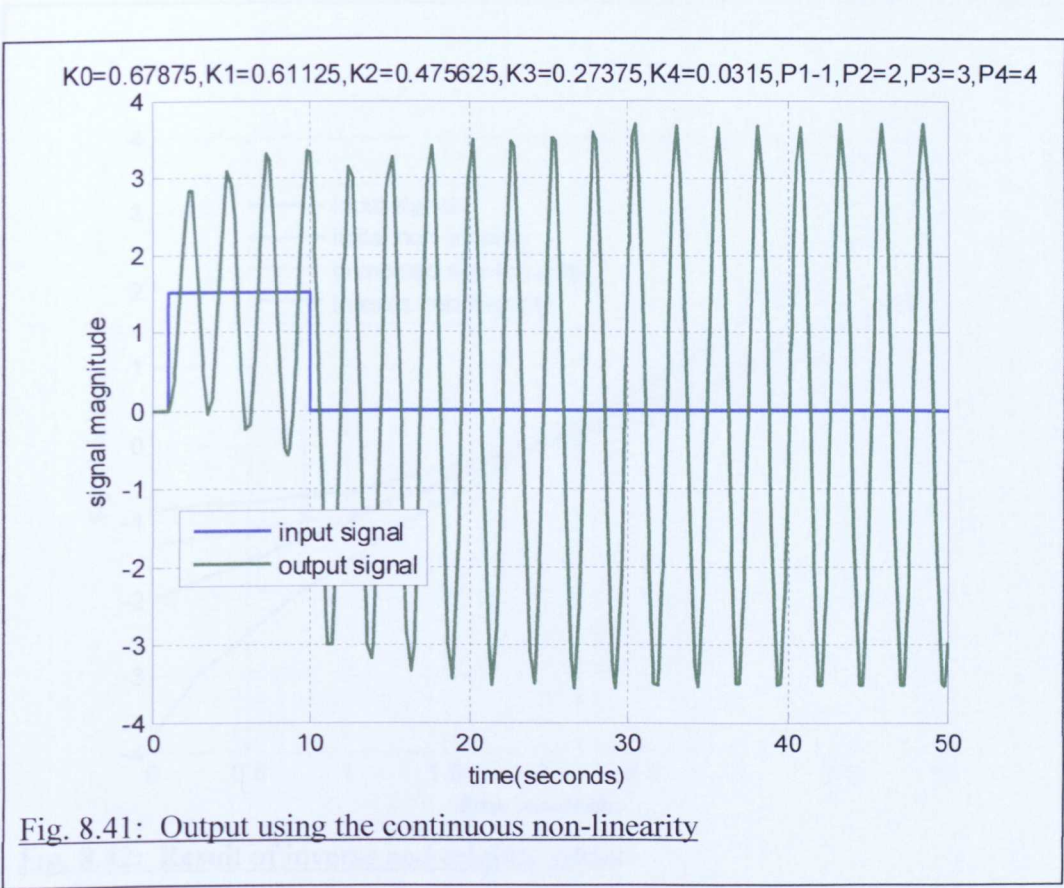
→

Table 8.5

Inverse of table 8.4

1/ Δy	x	y
	2.0715	5
31.746	2.04	4
3.6530	1.76625	3
2.0997	1.29	2
1.6360	0.67875	1
1.4733	0.0000	0
1.6360	-0.67875	-1
2.0997	-1.29	-2
3.6530	-1.76625	-3
31.746	-2.04	-4
	-2.04375	-5





From Figure 8.41 the magnitude of the limit-cycle was found to be 3.57 ± 0.2 . Looking at the describing function, Figure 8.39 for this polynomial approximation, the limit-cycle predicted magnitude was 3.65 ± 0.05 .

When looking at the characteristics of this non-linearity, and its inverse, Figure 8.40, it was clear that the inverse did not cover the extremities of the original non-linearity and so could not create a full inverse. This would be a feature of all systems in which saturation plays a part. It was not possible to overcome the physical boundary limitations of a system. However, within those boundaries the inverse process was very effective as could be seen in Figure 8.42.

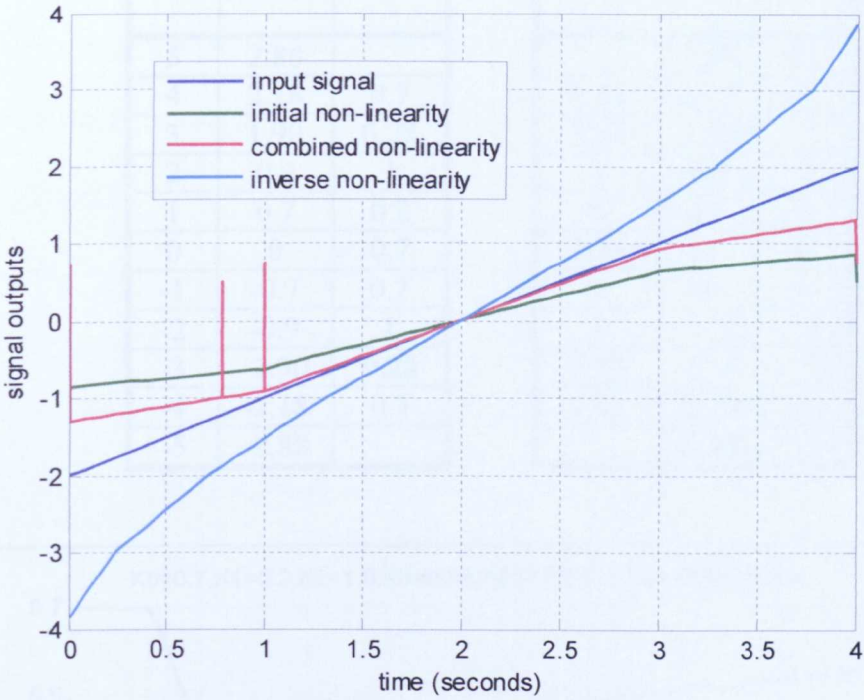


Fig. 8.42: Result of inverse and original signal

8.6.3.2 A continuous polynomial which doesn't exhibit saturation

The second polynomial did not exhibit saturation and therefore would be capable of having a perfect inverse. The polynomial used was

$$p = -0.0001x^9 - 0.0032x^7 + 0.0501x^5 - 0.2726x^3 + 0.9256x$$

It was approximated by the data in Table 8.6

Table 8.6

Ninth-order Polynomial Data

x	y (p)	Δy
5	2.88	
4	2.18	0.7
3	1.90	0.28
2	0.9	1
1	0.7	0.2
0	0	0.7
-1	-0.7	0.2
-2	-0.9	1
-3	-1.90	0.28
-4	-2.18	0.7
-5	-2.88	

Table 8.7

Inverse of Table 8.6

Slope	x	y
	2.88	5
1.43	2.18	4
3.57	1.90	3
1	0.9	2
5	0.7	1
1.43	0	0
5	-0.7	-1
1	-0.9	-2
3.57	-1.90	-3
1.43	-2.18	-4
	-2.88	-5

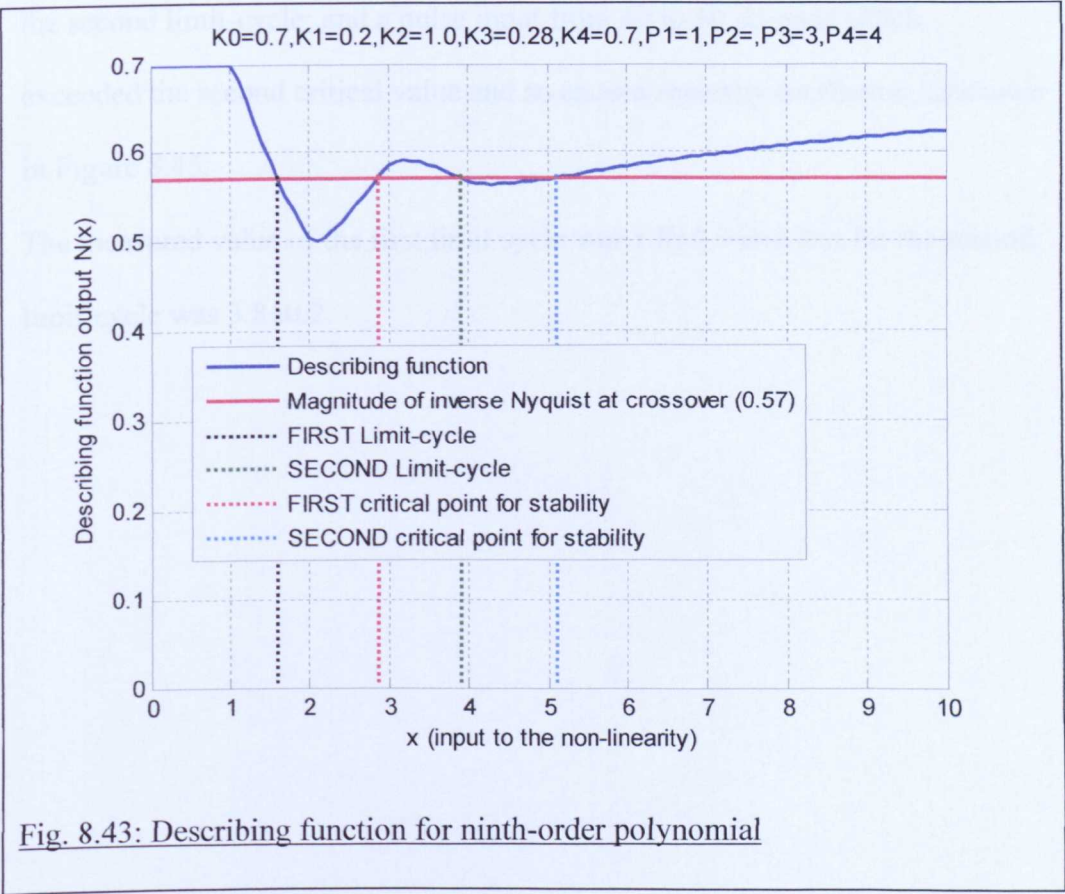


Fig. 8.43: Describing function for ninth-order polynomial

From Figure 8.39 there was a first limit-cycle at 1.6 ± 0.05 , a second limit-cycle at 3.9 ± 0.07 ; a first critical point at 2.88 ± 0.05 and a second critical point at 5.1 ± 0.07 .

The fuzzy non-linearity and its inverse were the next items created, Figure 8.44. Then followed the test of a ramp signal being input to the open-loop arrangement of the non-linearity in series with its inverse. This gave the result shown in Figure 8.46. The actual SIMULINK arrangements that were used were as shown in Appendix 3.4.

There was a unit input from 1 to 10 seconds, sufficient to trigger the first limit - cycle; a pulse input of 4.5 units from of 20 to 30 seconds, sufficient to trigger the second limit-cycle; and a pulse input from 40 to 50 seconds which exceeded the second critical value and so caused runaway oscillation, as shown in Figure 8.45.

The measured value of the first limit cycle was 1.8 ± 0.3 and that for the second limit-cycle was 3.8 ± 0.2 .

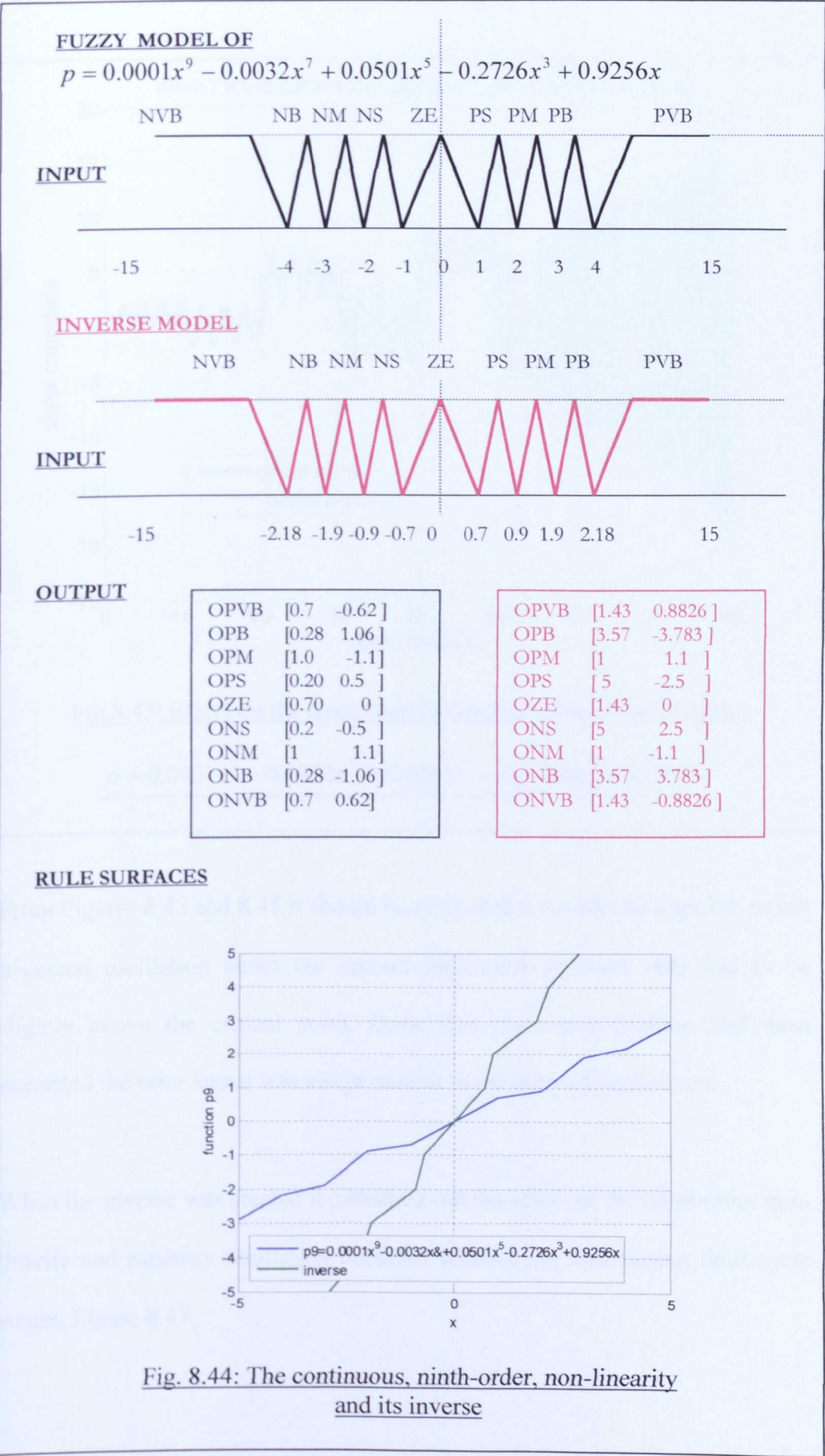
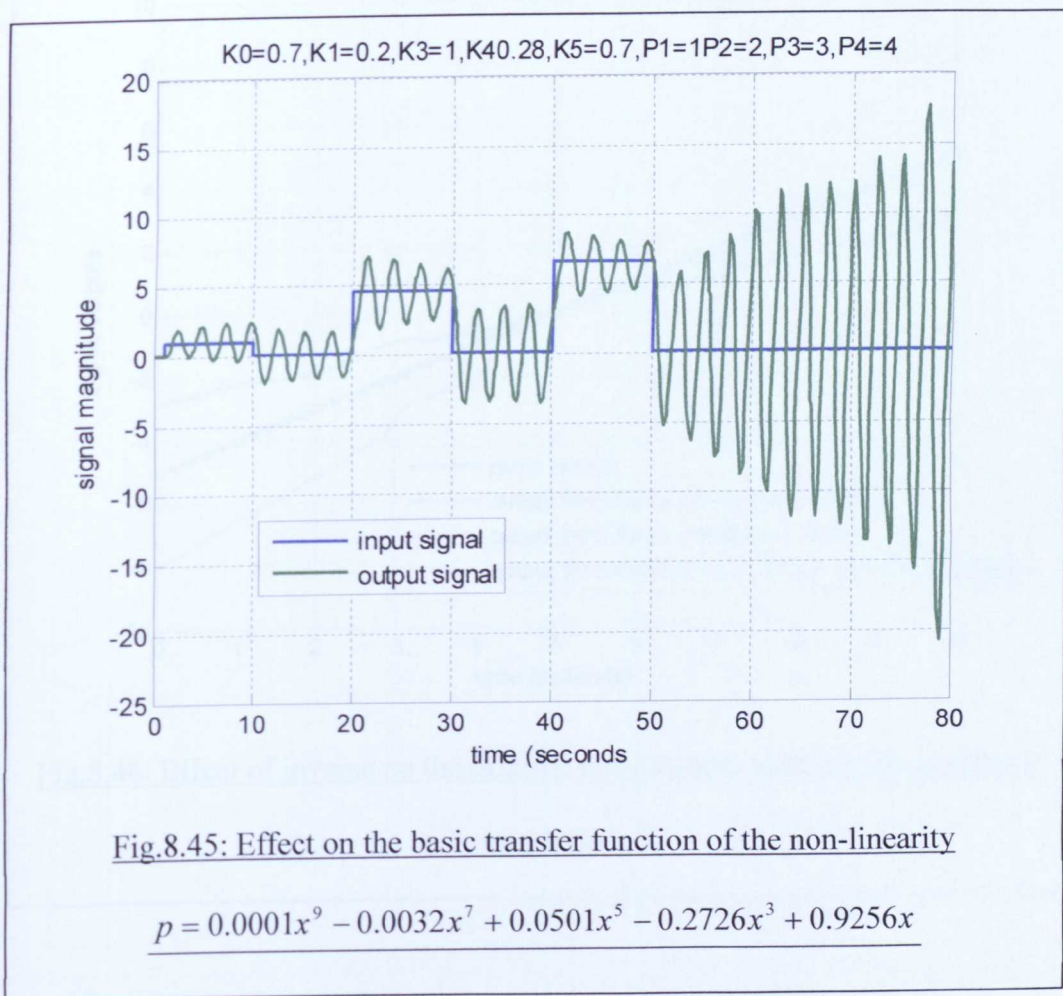


Fig. 8.44: The continuous, ninth-order, non-linearity and its inverse



From Figures 8.43 and 8.45 it should be noticed that the second impulse, which triggered oscillation about the second limit-cycle position only had to be slightly above the critical point. Once that cross-over position had been exceeded the error signal was swept around to the limit-cycle position.

When the inverse was created it cancelled out the effect of the ninth-order non-linearity and runaway oscillation occurred without any intervening limit-cycle stages, Figure 8.47.

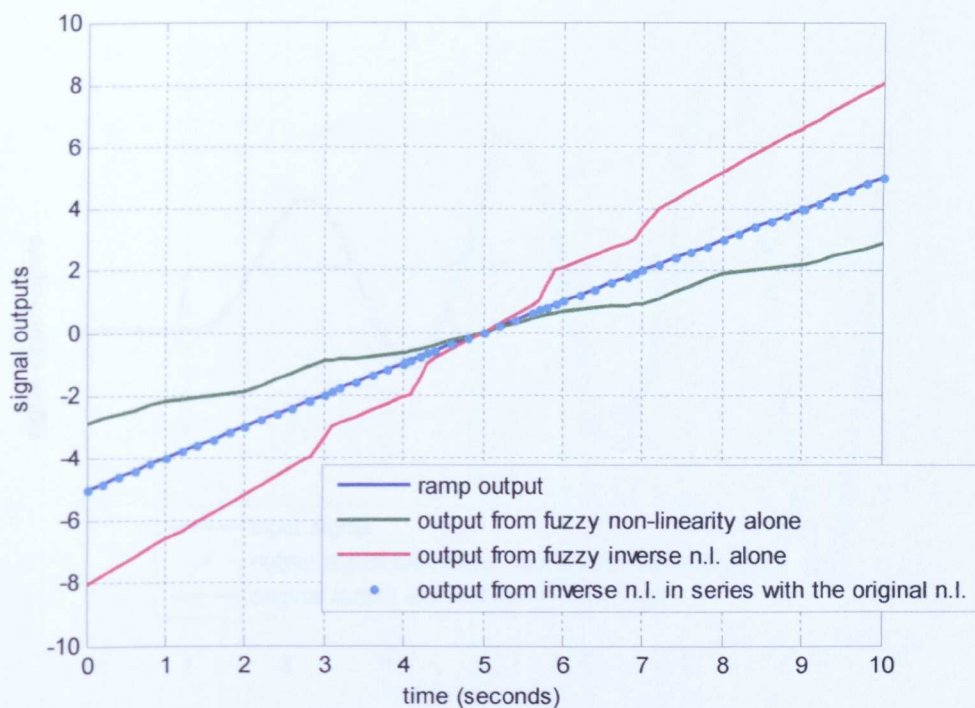


Fig.8.46: Effect of inverse on the original non-linearity with a ramp input and open-loop

8.7 Some general comments

In this chapter it has been demonstrated how effective fuzzy logic can be in compensating for the effects of many non-linearities and in stabilising systems that are otherwise unstable. It has also been demonstrated that periodic inverse models can be used to compensate for non-linearities in a system. The results of the simulations show that the fuzzy logic controller can be used to compensate for non-linearities in a system and that the system can be stabilised. The results of the simulations also show that the fuzzy logic controller can be used to compensate for non-linearities in a system and that the system can be stabilised. The results of the simulations also show that the fuzzy logic controller can be used to compensate for non-linearities in a system and that the system can be stabilised.

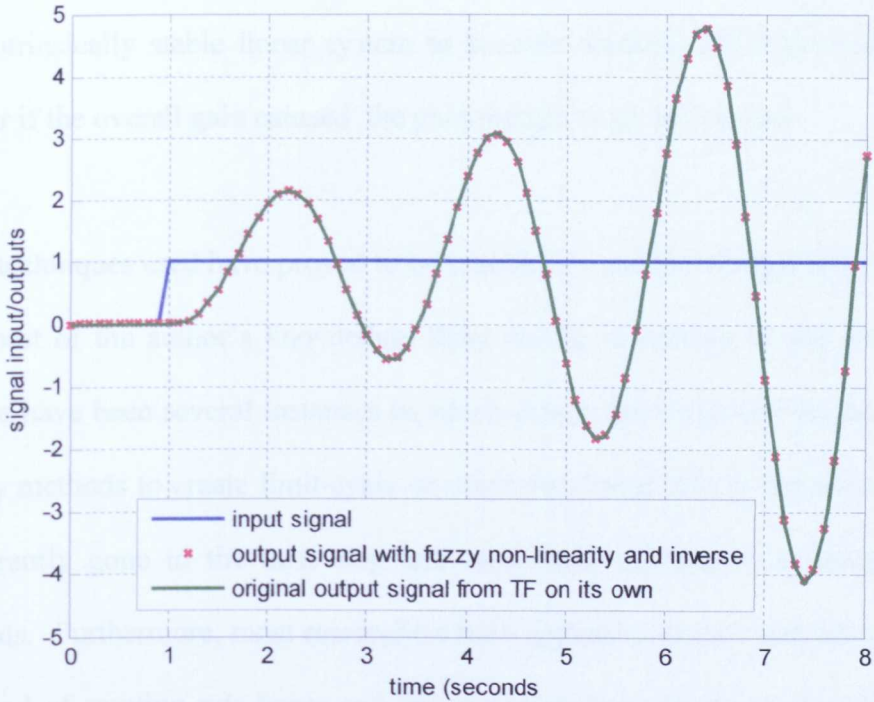


Fig. 8.47: Inverse in series with the 9th-order non-linearity in series with the standard test transfer function and unit feedback compared with the performance of the test transfer function on its own.

8.7 Some general comments

In this chapter it has been demonstrated that inverses could be created and so the effects of many non-linearities could be erased from a system. It has also been demonstrated that partial inverses could equally easily be devised so that non-linear characteristics could be selectively modified. The transfer function chosen to represent the test system was a third-order system with gain set sufficiently high that it gave an unstable response. The system was tested with a positive gain margin and again the insertion of a non-linearity in series with the transfer function gave a limit-cycle response but only if the inherent gain in the non-linearity was raised so that the gain margin was reduced to zero. If

the inverse non-linearity was miss-calculated then it was possible for it to cause an intrinsically stable linear system to become unstable but this would only occur if the overall gain caused the gain margin to go below zero

The techniques used have proved to be remarkably simple to apply although, to the best of the author's knowledge, there are no references in the literature. There have been several instances in which papers have reported on the use of fuzzy methods to create limit-cycle or other non-linear effects but nobody has apparently gone to the next step and used these techniques to modify the signals. Furthermore, most researchers have appeared to stay with Mamdani's method of creating rule-bases and this approach does not easily lend itself to design work. One surprising feature was that, having found a method which enables non-linear design and system-modification to be performed, there was considerable leeway in actually choosing the parameters necessary to complete the task.

Finally, the first of the two examples in section 8.6.3 was that of a continuous non-linear system which had an inherent saturation characteristic. All systems ultimately possess this characteristic and it represents a real physical limit beyond which nothing can respond. Beyond the saturation limit there was no signal. The techniques presented here are techniques for modifying and adjusting signals which exist. They cannot create a signal when there is nothing there. If they did then the newly-formed signal would be pure invention and might bear no relation to reality.

8.8 Conclusions

A considerably amount of work has been done by many researchers in trying to understand non-linear effects from an analytical standpoint (Nijmeijer and van der Shaft, 1990). Such work has not paid sufficient heed to the fact that non-linearities could inherently most easily be explained geometrically. In this chapter the whole concept of inverse non-linearities has been explored from a geometrical viewpoint and it was found that, basically, a inverse non-linear function was simply a reflection of the original non-linearity about the 45° line. Using this notion it has been possible to produce a general algorithm which enabled inverse non-linearities to be designed for any non-linear system which did not possess memory and so had a real describing function. However, all of this work relies on the assumption that the describing function approach always correctly predicts limit-cycle behaviour – it usually does but there are exceptions (Engelberg, 2002).

Complete inverse non-linear functions were not always required and by using describing functions as reference-levels it has proved possible to design partial inverse functions which enabled undesirable effects to be removed but also built-in other desirable features. An example would be the ability to make an inherently unstable system stable.

Chapter Nine

Conclusions and Suggested Further Work

9.1 Conclusions

Two disparate subject areas have been brought together in this research and the result has been the development of new techniques which enable non-linear effects to be studied and signal-modification to take place in a manner which has not previously been available. To set the scene, after the general introduction, chapter two was devoted to a description of the difference in characteristics between linear and non-linear systems and a description of the more usual types of non-linearity. Reference was continually made to these 'standard-forms' throughout the thesis as the various techniques of analysis and design were introduced. Gradually the various features associated with them were uncovered and patterns began to emerge. Patterns which ultimately aided the design and modification of non-linear systems in general.

The next stage was to decide which analytical techniques might be most helpful in the investigation. Of all the techniques available only two provided approaches which were at all tractable and these were both graphical: the phase-plane approach and the describing function method. Of the two, the describing function approach offered the greater potential for analysing and, ultimately, modifying non-linear systems. So that became the method of choice.

Stability analysis was discussed and a method from the earliest days of using describing functions turned out to be the most useful by far for predicting when non-linear systems would become unstable: the Kochenburger Stability Criterion which linked Nyquist and describing function techniques. This criterion could be used with Nyquist or Inverse Nyquist loci. The inverse locus was more convenient and was the locus used throughout this work. A condensed method of using Kochenburger's criterion, shown in Figure 3.5, was chosen as it allowed easy measurements to be made and emerging patterns of behaviour to be seen. This technique was used repeatedly, for each new describing function obtained, to predict the magnitudes of limit-cycles and where critical points existed.

The standard graphical method for obtaining describing functions was explained in detail and then a general solution, together with an associated implementation algorithm, was obtained in chapter four. The development of this particular general method for the family of real, as opposed to complex, describing functions was considered a necessity so that the whole group of such functions could be obtained relatively easily and quickly. With the availability of this facility it became possible to see patterns emerging between groups of non-linearities. To the best of the author's knowledge, the existence of such relationships has not appeared in the literature. Certainly, the general method for obtaining real describing functions and the associated algorithm has not appeared in the literature.

Chapter six presented a condensed survey of how fuzzy systems of various types could be used in this investigation. It was not meant to be a detailed explanation of how fuzzy controllers work, rather how they can be used. In fact a couple of fuzzy systems which the author thinks have potential for signal analysis and design have been described even though they were subsequently postponed to the further work category.

The meat of the investigation started in chapter seven with the simulation of non-linear systems by fuzzy means. Again, the standard non-linearities were considered, the condensed version of Kochenburger's method was used to predict the size of limit-cycles and then the results obtained with fuzzy non-linearities compared with them. Once the technique had been proved by comparing fuzzy non-linearities with standards available in the Matlab toolbox, more sophisticated systems were devised which theory predicted should produce multiple limit cycles. The Matlab fuzzy toolbox did not lend itself to very complicated designs and so the investigation was limited to two systems, the three and four breakpoint cases, each of which produced double limit-cycles. Using the describing function and the inverse-Nyquist locus the positions of the limit-cycles were predicted as well as critical cross-over points. The practical results agreed very closely with the predictions. Some extra features were also observed which could be explained by reference to the describing function/inverse Nyquist diagrams.

The describing function/inverse Nyquist diagram rapidly proved its use as a design tool and it was used as the starting point for most of the inverse

functions produced in chapter eight. Families of mutual inverse non-linear systems were predicted and successfully produced. Further, partial inverses were also produced which enable various features to be turned on or off. For instance, making an inherently unstable system stable or vice-versa; also switching off one or the other limit-cycles in two-cycle systems.

The investigation had been limited to real systems which did not possess memory and which did not produce complex describing functions. The reason for this self-imposed restriction was because of the very high computational demands of the Matlab fuzzy logic toolbox. It was found that with more than one input to the fuzzy logic toolbox the system response was far too slow. This ruled out many interesting systems such as those involving backlash, hysteresis of any sort and the jump-resonance cases. However, it was felt that if real systems were initially investigated and these were thoroughly understood then the more sophisticated cases could be considered in later work. Detailed study of the real systems in this research proved productive, enabling useful results and conclusions to be drawn.

9.2 Suggested further work

Although a potential strength of fuzzy set theory lies in its algorithmic promise there is a wall which the subject must surmount: that of the rapid increase in computational demand as the complexity of the problems increase. The theory is not yet computationally able to solve large and complex problems in an efficient manner. For this reason considerable research effort has gone into the development of fuzzy algorithms but, so far, to little published success. There

has been recent work in developing scripting languages (e.g.: FuzzScript¹) for the subject. Although the results so far appear limited, this development work is being attempted by an army of enthusiasts under the umbrella of The Code Project. This acts as a development resource for software developers, mostly using Microsoft resources, who are able to use published material under the Code Project Open Licence (CPOL)². Whether this development will aid fundamental research in the subject remains to be seen. Several commercial tools are available but they tend to be expensive and dedicated to specific tasks. A major effort by the professional research community is the development of the Fuzzy Markup Language (FML) (Acampora and Loia, 2005) and this is currently being standardized by a task force within the IEEE CIS Standards Committee. At present it is probably better to develop one's own software in-house.

The development of design tools in-house, tailored to our specific research needs, should make the investigation of devices such as the inference filter and the positive-negative rule scenario much easier to handle. The inference filter, or its counterpart the torque defuzzifier hold out the possibility of much finer control of the signal modification process. The positive-negative rule scenario can be generalised to a set of n parallel devices for which the hyperinference engine acts like a programmable switching device. In this form it effectively becomes a supervisory controller (Linkens *et al.*, 1993) which can select the best rule scenario for the conditions at hand.

¹ A fuzzy logic control language – see Carmel Gafa (2009)- CPOL licence

² The code project – www.codeproject.com – an open site for programming enthusiasts.

In this research the rapid increase in computational complexity as the size of the fuzzy control machine increases is the main reason why some of the more sophisticated devices were earmarked as future work. The investigation of systems with two or more inputs is the logical next step so that non-linear systems with memory, eg.: hysteresis, can be brought into the basic framework that has already been developed. This will mean investigating complex, as opposed to real, describing functions. This will, in turn, raise the possibility of modifying the behaviour of limit-cycle oscillations in ways which change their frequency of oscillation as well as their amplitude. Another non-linearity which should be amenable to the approach used in this thesis is the phenomena of jump-resonance; the possibility exists of being able to selectively change the characteristic of a system from the hard-spring to the soft-spring case. Another feature which deserves investigation is the hint, encountered on a few occasions during the present research, that the higher limit-cycles sometimes exhibit amplitude modulation.

The work done so far has demonstrated how the undesirable effects of simple non-linearities can be cancelled out, or at least modified, so that for the range of parameters in which the system is operating a more linear behaviour reigns. As mentioned earlier the received wisdom was it would be impossible to find complete inverses. That, it thought, would be the equivalent of finding an inverse for the state matrix of the system and was considered impossible since the principle of superposition does not apply. However, although the previous statements may be true for analytical solutions the fuzzy logic approach is

essentially geometric. Using the fuzzy approach, the problem reduces to one of finding the inverses of the rule-surface. For a two-input system, the minimum if a complex is to be investigated, requires the finding of an inverse of a 2-dimensional surface which should not be too difficult. Systems with more than two inputs will require higher-order geometries. In such cases the system-modification abilities of the fuzzy controller would be really tested.

The work presented here has all been simulation and using only one transfer function for the linear element. Other transfer functions were tested but to a much lesser extent. The system appears to be robust as far as the design of the non-linear inverses was concerned. What needs to be investigated further is the sensitivity of the technique to parameter and structural changes of real non-linear systems. Furthermore, there may be other approaches, besides the fuzzy approach, which might be useable in creating non-linear inverses but, as far as the author has seen in the literature, although the creation of limit-cycle behaviour has been reported nobody has followed through with attempts to develop general methods for the design of non-linear inverses.

Ultimately, if presented with a non-linear plant whose characteristics are unknown then it would be desirable to be able to design a self-organising fuzzy controller (Mamdani, 1979; Tsang, 2001) which could modify the non-linear behaviour automatically.

The whole area of signal and system conditioning by fuzzy means is one which has hardly been investigated.

REFERENCES

- Acampora, G. and Loia, V., (2005), *Fuzzy Inoperability and Scalability for Adaptive Domotic Framework*, *IEEE Trans. Industrial Informatics*, Vol. 1, No. 2, May 2005, pp 97-111.
- Aracil, J. and Gordillo, F., (2004), *Describing function method for stability analysis of PD and PI fuzzy controllers*, *Fuzzy Sets and Systems*, Vol. 143, pp 233-249.
- Arcak, M., Larsen, M. and Kokotović, P., (2003), *Circle and Popov Criteria as Tools for Nonlinear Feedback Design*, *Automatica*, Vol. 39, No. 4, April, pp 643-650.
- Atherton, D.P., (1975), *Nonlinear Control Engineering*, Van Nostrand.
- Atherton, D.P. and Dorrah, H.T., (1980), *A survey on non-linear oscillations*, *Int. J. Control*, Vol. 31, pp 1041-1105.
- Atherton, D.P., (1981), *Stability of Non-Linear Systems*, Research Studies Press, Wiley.
- Atherton, D.P., (1996), *Early Developments in Nonlinear Control*, *IEEE Control Systems Magazine*, Vol. 16, Issue 3, pp 34-43.
- Bendixson, I., (1901), *Sur les courbes definies par des equations differentielles*, *Acta Mathematica*, Vol.24, pp 1-88.
- Branson, J.S. and Lilly, J.H., (2001), *Incorporation, Characterization and Conversion of Negative Rules into Fuzzy Inference Systems*, *IEEE Trans. Fuzzy Systems*, Vol. 9, No. 2, pp 253-268.
- Casillas, J., Cordon, O., del Jesus, M.J. and Herrera, F., (2005), *Genetic Tuning of Fuzzy Rule Deep Structures*, *IEEE Trans. Fuzzy Systems*, Vol 13, No.1, pp 13-29.

- Castro, J.L., (1995), *Fuzzy Logic Controllers are Universal Approximators*, IEEE Trans. Syst. Man & Cybernetics, Vol. 25, No. 4, April, pp 629-635.
- Cuesta, F., Gordillo, F., Aracil, J. and Ollero, A., (1999), *Stability Analysis of Nonlinear Multivariable Takagi-Sugeno Fuzzy Control Systems*, IEEE Trans. Fuzzy Systems, Vol. 7, No. 5, pp 508-520.
- Dutilh, J., (1950), *Theorie des servomechanism a relais*, Onde Elec., pp 438-445.
- Dutton, K., Thompson, S. and Barraclough, B., (1997), *The Art of Control Engineering*, Prentice-Hall.
- Engelberg, S., (2002), *Limitations of the Describing Function for Limit Cycle Prediction*, IEEE Trans Automatic Control, Vol. 47, No. 11, pp 1887-1890.
- Foo, T.J., (2001), *Stable Model Reference Adaptive Fuzzy Control of a Class of Nonlinear Systems*, IEEE Trans. Fuzzy Systems, Vol. 9, No. 4, pp 624-636.
- Gibbs, W., (1928), *Collected Works*, Longmans, New York.
- Gibson, J.E., (1963), *Nonlinear Automatic Control*, McGraw-Hill, New York, pp 405-410.
- Gordillo, F., Aracil, J. and Alamo, T., (1997), *Determining limit-cycles in fuzzy control systems*, Proc. IEEE Int. Conf. Fuzzy Systems., Barcellona, Spain, pp. 193-198.
- Grensted, P.E.W., (1955), *The Frequency-Response Analysis of Non-Linear Systems*, Proc.IEE, Vol. 102c.
- Güler, I. & Übeyli, E. D., (2005), *Adaptive neuro-fuzzy inference system for classification of EEG signals using wavelet coefficients*, J. Neuroscience Methods, Elsevier, Vol. 148, Issue 2, pp 113-121.
- Hu, B.G., Mann, G.K.L. and Gosine, R.G., (2001), *A systematic study of PID controllers – a function-based evaluation approach*, IEEE Trans. Fuzzy Systems, Vol. 9, No. 5, pp 699-712.

- Johnson, E.C., (1952), *Sinusoidal Analysis of Feedback Control Systems Containing Non-Linear Elements*, Trans. AIEE, Vol. 71, part II, pp 169-181.
- Kandel, A., (1982), *Fuzzy Techniques in Pattern Recognition*, Wiley, New York
- Kang, H.W., Cho, Y.S. and Youn, D.H., (1999), *On compensating nonlinear distortions of an OFDM system using an efficient adaptive predistorter*, IEEE Trans. Communications, Vol. 47, Issue 4, pp 522-526.
- Khalil, H. K., (2002), *Nonlinear Systems 3rd Ed.*, Prentice-Hall, pp 54-59, 280-288.
- Kickert, W.J.M. and Mamdani, E.H., (1978), *Analysis of a fuzzy logic controller*, Fuzzy Sets & Systems, Vol. 1, pp 29-44.
- Kiendl, H., (1994), *The Inference Filter*, Proc. EUFIT 94, Aachen, Vol. 1, pp 438-447.
- Kiendl, H., (1998), *Next Generation Fuzzy Systems*, Proc. EUFIT 98, Aachen, Vol. 2, pp 779-788.
- Kim, E., Lee, H. and Park, M., (2000), *Limit Cycle Prediction of a Fuzzy Control System Based on a Describing Function Method*, IEEE Trans. Fuzzy Systems, Vol. 8, No. 1, pp 11-22
- Kochenburger, R. J., (1950), *A frequency response method for analysing and synthesising contactor servomechanisms*, Trans. AIEE, Vol. 69, pp. 270-283.
- Krone, A. and Schwane, U., (1996), *Generating Fuzzy Rules from Contradictory Data of Different Control Srearegies and Control Performances*, Prof. FUZZ-IEEE '96m, pp 492-497, New Orleans.
- Kwok, D.P., Tan, P., Li, C.K. & Wang, P., (1991), *Analysis and Design of Fuzzy PID Control Systems*, Hong Kong Polytechnic, Control 91, Int. Conf. Control, ieeexplore.ieee.org.

- LaSalle, J.P., and Lefschetz, S., (1961), *Stability by Liapunov's direct method with applications*, Academic Press, New York,
- Lee, H.J., Park, J.B. and Chen, G., (2001), *Robust Fuzzy Control of Nonlinear Systems with Parametric Uncertainties*, IEEE Trans. Fuzzy Systems, Vol. 9, No. 2, pp 369-379.
- Leigh, J.R., (1983), *Essentials of Non-Linear Control Theory*, Peter Peregrinus, London,
- Liapunov, A., (1961), *Stability in nonlinear control systems*, Princeton University Press, N.J. [*English translation of his original paper of 1892*]
- Lim, Y.H., Cho, Y.S., Cha, I.W. and Youn, D.H., (2002), *An adaptive nonlinear prefilter for compensation of distortion in nonlinear systems*, IEEE Trans. Signal Processing, Vol. 46, Issue 6, pp 1726-1730.
- Liming Hu, Cheng, H.D. & Zhang, M., (2007), *A high performance edge detector based on fuzzy inference rules*, Information Sciences, Elsevier, Vol. 177, issue 21, pp 4668-4784.
- Linkens, D.A. and Abbod, M.F., (1993), *Supervisory Intelligent Control Using a Fuzzy Logic Hierarchy*, Trans. Inst.M.C., Vol. 15, No. 3, pp 112-132.
- Loab, J.M., (1956), *Recent Advances in Nonlinear Servo Theory*, in Oldenburger R., ed., *Frequency Response*, Macmillan, New York.
- Lur'e, A.L., (1957), *Some nonlinear problems in the theory of automatic control*, H.M.S.O., London.
- Mamdani, E.H.,(1974) *Applications of fuzzy algorithm for simple dynamic plant*, Proc. IEEE, Vol. 121, No. 12, pp 1585-1588.
- Mamdani, E.H. and Assilian, S., (1975), *An experiment in linguistic synthesis with a fuzzy logic controller*, International Journal of Man-Machine Studies, Vol. 7, No 1, pp. 1-13.

- Mamdani, E.H. and Procyk, T.J., (1979), *A Linguistic Self-Organizing Controller*, Automatica, Vol. 15, Issue 1, pp 15-30.
- Mansoor, E.G., Zolghadri, M.J. and Katebi, S. D., (2007), *A Weighting Function for Improving Fuzzy Classification system performance*, Fuzzy Sets and Systems, Vol. 158, Issue 5, pp 583-591.
- Mitaim, S. and Kosko, B., (2001), *The Shape of Fuzzy Sets in Adaptive Function Approximation*, IEEE Trans. Fuzzy Systems, Vol. 9, No. 4, pp 637-656.
- Nagrath, I.J. and Gopal, M., (1986), *Control Systems Engineering 2nd Ed.*, Wiley, pp. 625-640.
- Nassirharand, A. and Karimi, H., (2006), *Nonlinear controller synthesis based on inverse describing function technique in the MATLAB environment*, Advances in Engineering Software, Vol. 37, Issue 6, pp 370-374.
- Nijmeijer, H. and van der Schaft, (1990), *Nonlinear dynamical control systems*, Springer-Verlag, New York.
- Nyquist, H., (1932), *Regeneration Theory*, Bell System Technical Journal,
- Page, G. F., Gomm, J.B. and Rapp, A., (1996), *Fuzzy Logic PID Systems*, International Automatisierungssymposium, Wismar, Germany.
- Perng, J-W., Wu, B-F. Lee, T-T., (2003), *Gain-Phase Margin Analysis of Fuzzy Control Systems*, Proc. IEEE Conference on Fuzzy Systems, Taiwan, pp 3681-3686.
- Passino, K.M. and Yurkowich, S., (1997), *Fuzzy Control*, Addison-Wesley.
- Popov, V.M., (1960), *Criterion of quality for nonlinear controlled systems*, First IFAC World Congress, Moscow, pp 173-176.
- Ross, T., (2004), *Fuzzy Logic with Engineering Applications 2nd Ed.*, McGraw-Hill, pp 90-95.

- Sepulveda, R., Castillo, O., Melin, P., Rodriguez-Diaz, A. and Montiel, O., (2007), *Experimental Study of intelligent controllers under uncertainty using type 1 and type 2 fuzzy logic*, Information Sciences, Vol. 177, Issue, 10, May, pp 2023-2048.
- Slotine, J.-J. and Li, W., (1991), *Applied non-Linear Control*, Prentice-Hall. N.J.
- Smith, R.C., Bouron, C. and Zrostlik, (2000), *Partial and full inverse compensation for hysteresis in smart material systems*, Proc. of American Control Conf., Vol. 4, pp 2750-2754.
- Steeb, W-H., (2002), *The Non-Linear Workbook 2nd Ed.*, World Scientific, pp 211-245.
- Sugeno, M., (1985), *Industrial Applications of Fuzzy Control*, Elsevier Science Pub. Co.
- Takagi, H., (1993), Survey: *Fuzzy logic applications of image-processing equipment*, in Industrial Applications of Fuzzy Control and Intelligent Systems, e John Yen & Reza Langari, Van Nostrand Reinhold.
- Tewari, H., (2002), *Modern Control Design*, Wiley& Sons.
- Tsang, K.M., (2001), *Auto-tuning of Fuzzy Logic Controllers for Self-Regulating Processes*, Fuzzy Sets and Systems, Vol. 120, Issue 1, pp 169-179.
- Tustin, A., (1947), *The effects of backlash and of speed dependent friction on the stability of closed cycle control systems*, JIEE, part II, Vol. 94, pp 143-151.
- Vashkov, G. and Fukuda, T., (1998), *Fuzzy Control Based on Self-Tuning Inverse Fuzzy Model*, Proceedings 6th European Congress on Intelligent Techniques & Soft Computing, Aachen, Germany, pp 839-843.
- Wang, C.-H., Li, G., Hong, T.-P. and Tseng, S.-S., (1998), *Integrating Fuzzy Knowledge by Genetic Algorithms*, IEEE Trans. Evolutionary Computation, Vol. 2, No. 4, pp 138-149.

West, J.C. and Douce, J.L., (1954), *The Frequency Response of a Certain Class of Nonlinear Feedback Systems*, Br. J. Appl. Phys., Vol. 5.

Zadeh, L.A., (1965), *Fuzzy Sets*, Information and Control , Vol. 8, pp 338-353.

Zadeh, L.A., (1969), *Fuzzy Algorithms*, Information and Control, Vol. 19, pp94-102.

Zames, G. and Falb, J.L., (1968), *Stability conditions for systems with monotone and slope-restricted nonlinearities*, SIAM Journal of Control and Optimization, Vol. 6, pp 89-108.

Zimmermann, H. -J., (1996), *Fuzzy Set Theory*, 3rd Ed., Kluwer, Germany.

APPENDIX A1

**DESCRIBING FUNCTIONS OF REAL, NOT COMPLEX,
SYSTEMS**

APPENDIX A1.1

A GENERAL SOLUTION FOR A FAMILY OF REAL, SINGLE-VALUED, DESCRIBING FUNCTIONS

For a family of single-valued describing functions, which can be approximated to straight lines between the points (break-points) at which their slopes change value, Gibson's graphical method can be extended. Referring to Figures 4.2 & 4.3, for n breakpoints occurring at horizontal positions $P_0 P_1 \dots P_i \dots P_n$, slopes $K_0 K_1 \dots K_i \dots K_n$, jumps in the vertical plane at $Q_1 Q_2 \dots$ and angles on the sinusoidal input, where the breakpoints occur, of $\alpha_1 \alpha_2 \dots \alpha_i \dots \alpha_n, \pi/2$, the describing function will have the form:

$$A_1 = \frac{4}{\pi} \left[\int_0^{\alpha_1} y \cdot \sin(\omega t) \cdot d(\omega t) + \int_{\alpha_1}^{\alpha_2} y \cdot \sin(\omega t) \cdot d(\omega t) + \dots \right. \\ \left. \dots + \int_{\alpha_{i-1}}^{\alpha_i} y \cdot \sin(\omega t) \cdot d(\omega t) + \dots + \int_{\alpha_n}^{\frac{\pi}{2}} y \cdot \sin(\omega t) \cdot d(\omega t) \right] \dots \dots \dots (4.3)$$

with $y = K_i x$ and $x = X \cdot \sin(\omega t)$

Between $0 \leq \omega t < \alpha_1$ $y = K_0 x$

Between $\alpha_1 \leq \omega t < \alpha_2$ $y = mx + c = K_1 x + P_1(K_0 - K_1)$

Between $\alpha_2 \leq \omega t < \alpha_3$ $y = K_2 x + P_1(K_0 - K_1) + P_2(K_1 - K_2)$

...

Between $\alpha_i \leq \omega t < \alpha_{i+1}$ $y = K_i x + P_1(K_0 - K_1) + \dots + P_i(K_{i-1} - K_i)$

...

And between $\alpha_n \leq \omega t < \frac{\pi}{2}$

$$y = K_n x + P_1(K_0 - K_1) + \dots + P_i(K_{i-1} - K_i) + \dots + P_n(K_{n-1} - K_n)$$

$$\therefore A = \frac{4}{\pi} \left[\int_0^{\alpha_1} K_0 x \sin(\omega t).d(\omega t) + \int_{\alpha_1}^{\alpha_2} [K_1 x + P_1(K_0 - K_1)] \sin(\omega t).d(\omega t) + \right. \\ \left. \int_{\alpha_2}^{\alpha_3} [K_2 x + P_1(K_0 - K_1) + P_2(K_1 - K_2)] \sin(\omega t).d(\omega t) + \dots \right. \\ \left. + \int_{\alpha_i}^{\alpha_{i+1}} [K_i x + P_1(K_0 - K_1) + \dots + P_i(K_{i-1} - K_i)] \sin(\omega t).d(\omega t) + \dots \right. \\ \left. + \int_{\alpha_n}^{\frac{\pi}{2}} [K_n x + P_1(K_0 - K_1) + \dots + P_i(K_{i-1} - K_i) + \dots + P_n(K_{n-1} - K_n)] \sin(\omega t).d(\omega t) \right]$$

For convenience, let $A_i = A_{i1} + A_{i2}$ where, putting $x = X \sin(\omega t)$:

$$A_{i1} = \frac{4}{\pi} \left[\int_0^{\alpha_1} K_0 X \sin^2(\omega t).d(\omega t) + \dots + \int_{\alpha_i}^{\alpha_{i+1}} K_i X \sin^2(\omega t).d(\omega t) + \dots \right. \\ \left. + \int_{\alpha_n}^{\frac{\pi}{2}} K_n X \sin^2(\omega t).d(\omega t) \right]$$

and

$$A_{i2} = \frac{4}{\pi} \left[\int_{\alpha_1}^{\alpha_2} P_1(K_0 - K_1) \sin(\omega t).d(\omega t) + \right. \\ \left. \int_{\alpha_2}^{\alpha_3} [P_1(K_0 - K_1) + P_2(K_1 - K_2)] \sin(\omega t).d(\omega t) + \dots \right. \\ \left. + \int_{\alpha_i}^{\alpha_{i+1}} [P_1(K_0 - K_1) + \dots + P_i(K_{i-1} - K_i)] \sin(\omega t).d(\omega t) + \dots \right. \\ \left. + \int_{\alpha_n}^{\frac{\pi}{2}} [P_1(K_0 - K_1) + \dots + P_i(K_{i-1} - K_i) + \dots + P_n(K_{n-1} - K_n)] \sin(\omega t).d(\omega t) \right]$$

Substituting $\sin^2(\omega t) = \frac{1}{2} - \frac{1}{2} \cos(2\omega t)$ into A_{i1} gives

$$A_{i1} = \frac{4}{\pi} \left[\frac{1}{2} \int_0^{\alpha_1} K_0 X.d(\omega t) + \frac{1}{2} \int_{\alpha_1}^{\alpha_2} K_1 X.d(\omega t) + \dots + \frac{1}{2} \int_{\alpha_i}^{\alpha_{i+1}} K_i X.d(\omega t) + \dots \right. \\ \left. + \frac{1}{2} \int_{\alpha_n}^{\frac{\pi}{2}} K_n X.d(\omega t) - \frac{1}{2} \int_0^{\alpha_1} K_0 X.\cos(2\omega t).d(\omega t) - \frac{1}{2} \int_{\alpha_1}^{\alpha_2} K_1 X.\cos(2\omega t).d(\omega t) - \dots \right. \\ \left. - \frac{1}{2} \int_{\alpha_i}^{\alpha_{i+1}} K_i X.\cos(2\omega t).d(\omega t) - \dots - \frac{1}{2} \int_{\alpha_n}^{\frac{\pi}{2}} K_n X.\cos(2\omega t).d(\omega t) \right]$$

$$\begin{aligned}
 \therefore A_{11} &= \frac{4X}{2\pi} \left[K_0(\alpha_1 - 0) + K_1(\alpha_2 - \alpha_1) + \dots + K_i(\alpha_i - \alpha_{i-1}) + \dots + K_n \left(\frac{\pi}{2} - \alpha_n \right) \right. \\
 &\quad - \frac{K_0}{2} \{ \sin(2\alpha_1) - \sin(0) \} - \frac{K_1}{2} \{ \sin(2\alpha_2) - \sin(2\alpha_1) \} \\
 &\quad - \frac{K_2}{2} \{ \sin(2\alpha_3) - \sin(2\alpha_2) \} - \dots - \frac{K_i}{2} \{ \sin(2\alpha_{i+1}) - \sin(2\alpha_i) \} - \dots \\
 &\quad - \frac{K_n}{2} \{ \sin(\pi) - \sin(2\alpha_n) \} + \frac{(K_1 - K_0)}{2} \cdot \sin(2\alpha_1) + \dots + \frac{(K_i - K_{i-1})}{2} \cdot \sin(2\alpha_i) + \dots \\
 &\quad \left. + \frac{(K_n - K_{n-1})}{2} \cdot \sin(2\alpha_n) \right] \\
 &= \frac{2X}{\pi} \left[(K_0 - K_1)\alpha_1 + \dots + (K_{i-1} - K_i)\alpha_i + \dots + (K_{n-1} - K_n)\alpha_n + K_n \cdot \frac{\pi}{2} \right. \\
 &\quad \left. + (K_1 - K_0)\sin\alpha_1 \cdot \cos\alpha_1 + \dots + (K_i - K_{i-1})\sin\alpha_i \cdot \cos\alpha_i + \dots + (K_n - K_{n-1})\sin\alpha_n \cdot \cos\alpha_n \right] \\
 \therefore A_{11} &= \frac{2X}{\pi} \left[K_n \cdot \frac{\pi}{2} + \sum_{i=1}^n (K_{i-1} - K_i)\alpha_i - \sum_{i=1}^n (K_{i-1} - K_i)\sin\alpha_i \cdot \cos\alpha_i \right] \dots (A1.1.1)
 \end{aligned}$$

Looking at A_{12} :

$$\begin{aligned}
 A_{12} &= \frac{4}{\pi} \left[\int_{\alpha_1}^{\alpha_2} P_1(K_0 - K_1)\sin(\omega t) \cdot d(\omega t) + \right. \\
 &\quad \int_{\alpha_2}^{\alpha_3} [P_1(K_0 - K_1) + P_2(K_1 - K_2)]\sin(\omega t) \cdot d(\omega t) + \dots \\
 &\quad + \int_{\alpha_i}^{\alpha_{i+1}} [P_1(K_0 - K_1) + \dots + P_i(K_{i-1} - K_i)]\sin(\omega t) \cdot d(\omega t) + \dots \\
 &\quad \left. + \int_{\alpha_n}^{\frac{\pi}{2}} [P_1(K_0 - K_1) + \dots + P_i(K_{i-1} - K_i) + \dots + P_n(K_{n-1} - K_n)]\sin(\omega t) \cdot d(\omega t) \right] \\
 \therefore A_{12} &= \frac{4}{\pi} \left[P_1(K_0 - K_1)\cos\alpha_1 - P_1(K_0 - K_1)\cos\alpha_2 + \{P_1(K_0 - K_1) + P_2(K_1 - K_2)\}\cos\alpha_2 \right. \\
 &\quad - \{P_1(K_0 - K_1) + P_2(K_1 - K_2)\}\cos\alpha_3 \\
 &\quad + \{P_1(K_0 - K_1) + P_2(K_1 - K_2) + P_3(K_2 - K_3)\}\cos\alpha_3 - \dots \\
 &\quad - \{P_1(K_0 - K_1) + P_2(K_1 - K_2) + \dots + P_{i-1}(K_{i-2} - K_{i-1}) + P_i(K_{i-1} - K_i)\}\cos\alpha_i \\
 &\quad - \{P_1(K_0 - K_1) + P_2(K_1 - K_2) + \dots + P_i(K_{i-2} - K_{i-1}) + P_i(K_{i-1} - K_i)\}\cos\alpha_{i+1} + \dots \\
 &\quad + \{P_1(K_0 - K_1) + \dots + P_i(K_{i-1} - K_i) + \dots + P_n(K_{n-1} - K_n)\}\cos\alpha_n \\
 &\quad \left. - \{P_1(K_0 - K_1) + \dots + P_i(K_{i-1} - K_i) + \dots + P_n(K_{n-1} - K_n)\}\cos\frac{\pi}{2} \right]
 \end{aligned}$$

$$\therefore A_{12} = \frac{4}{\pi} [P_1(K_0 - K_1)\cos\alpha_1 + \dots + P_i(K_{i-1} - K_i)\cos\alpha_i + \dots + P_n(K_{n-1} - K_n)\cos\alpha_n]$$

$$\therefore A_{12} = \frac{4}{\pi} \left[\sum_{i=1}^n P_i(K_{i-1} - K_i)\cos\alpha_i \right] \dots\dots\dots (A1.1.2)$$

Now the general form of the describing function will be given by

$$N = \frac{A_{11} + A_{12}}{X}$$

$$\therefore N = \frac{2}{\pi} \left[K_n \cdot \frac{\pi}{2} + \sum_{i=1}^n (K_{i-1} - K_i)\alpha_i - \sum_{i=1}^n (K_{i-1} - K_i)\sin\alpha_i \cos\alpha_i \right. \\ \left. + \frac{2}{X} \sum_{i=1}^n P_i(K_{i-1} - K_i)\cos\alpha_i \right]$$

$$\therefore N = \frac{2}{\pi} \left[K_n \cdot \frac{\pi}{2} + \sum_{i=1}^n (K_{i-1} - K_i)\alpha_i - \sum_{i=1}^n (K_{i-1} - K_i)\sin\alpha_i \cos\alpha_i \right. \\ \left. + 2 \sum_{i=1}^n \frac{P_i}{X} (K_{i-1} - K_i)\cos\alpha_i \right]$$

since $\left(\frac{P_i}{X}\right) = \sin\alpha_i$, N becomes:

$$N = \frac{2}{\pi} \left[K_n \cdot \frac{\pi}{2} + \sum_{i=1}^n (K_{i-1} - K_i)\alpha_i + \sum_{i=1}^n (K_{i-1} - K_i)\sin\alpha_i \cos\alpha_i \right]$$

Also, since $\cos\alpha_i = \sqrt{1 - \sin^2\alpha_i}$

$$N = \frac{2}{\pi} \left[K_n \cdot \frac{\pi}{2} + \sum_{i=1}^n (K_{i-1} - K_i) \left(\sin^{-1}\left(\frac{P_i}{X}\right) + \left(\frac{P_i}{X}\right) \sqrt{1 - \left(\frac{P_i}{X}\right)^2} \right) \right] \dots (A1.1.3)$$

If Coulomb friction or relay action is present then the first term has to be adjusted. Referring to Figure 4.4, reproduced here for clarity, there is a relationship between P on the x (input) axis and Q on the y (output) axis:

$$K = \frac{Q}{P}$$

And as $P \rightarrow 0$ $\sin^{-1}\left(\frac{P}{X}\right) \rightarrow \sin^{-1}(0) \rightarrow 0$

$$\text{So } \frac{P}{X} \sqrt{1 - \left(\frac{P}{X}\right)^2} \rightarrow \frac{P}{X}$$

$$\therefore N = \frac{2}{\pi} \cdot K \cdot \frac{P}{X} = \frac{2}{\pi} \cdot \frac{Q}{P} \cdot \frac{P}{X} = \frac{2Q}{\pi X}$$

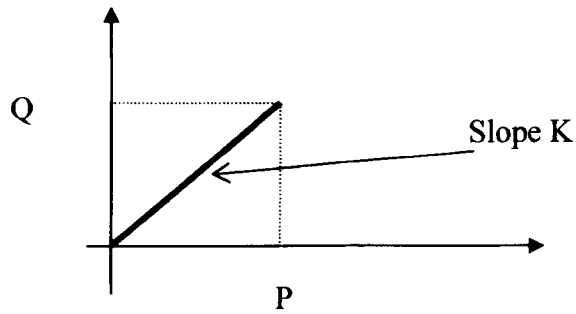


Fig. A1.1.1: Relationship between P and Q

APPENDIX 1.2

MATLAB CODE FOR THE GENERAL SOLUTION WITH FOUR BREAKPOINTS

Only the code for four breakpoints ($n=4$) has been shown because producing results for several different values of n simply leads to a lot of repetition.

```
%function [N] = polynomial approximation
%describing function of a non-linearity with an inner slope K0,
%a middle slopes K1, K2 and K3 and an outer slope K4.
%Break points at P1, P2, P3 and P4.
clear all
disp('This program plots out the describing function for')
disp('a polynomial approximation with FOUR break points')
disp('Please state the value of slope of the inner linear
section')
K0 = input('K0 = ')
disp('Please state the value of the slope in the first middle
linear section')
K1 = input('K1 = ')
disp('Please state the value of the slope of the next section')
K2 = input('K2 = ')
disp('Please state the value of the slope of the next section')
K3 = input('K3 = ')
disp('Please state the value of the slope of the outer linear
section')
K4 = input('K4 = ')
disp('Please state the value at which the first break-point
occurs')
P1 = input('P1 = ')
disp('Please state the value at which the second break-point
occurs')
P2 = input('P2 = ')
disp('Please state the value at which the third break-point
occurs')
P3 = input('P3 = ')
disp('Please state the value at which the fourth break-point
occurs')
P4 = input('P4 = ')
n=0;
for E=-5:0.1:5
    n=n+1;
    if E<=-P4
        N=(4/pi)*(K4*pi/4+((K0-K1)/2)*asin(-P1/E)+((K1-
K2)/2)*asin(-P2/E)+...
        +((K2-K3)/2)*asin(-P3/E)+((K3-K4)/2)*asin(-P4/E)...
        +((K0-K1)/2)*(-P1/E)*sqrt(1-(-P1/E)^2)+...
        +((K1-K2)/2)*(-P2/E)*sqrt(1-(-P2/E)^2)+...
        +((K2-K3)/2)*(-P3/E)*sqrt(1-(-P3/E)^2)+...
```

```

        + ((K3-K4)/2)*(-P4/E)*sqrt(1-(-P4/E)^2));
elseif E >=P4
    N=(4/pi)*(K4*pi/4+((K0-K1)/2)*asin(P1/E)+((K1-
K2)/2)*asin(P2/E)+...
    +((K2-K3)/2)*asin(P3/E)+((K3-K4)/2)*asin(P4/E)+...
    +((K0-K1)/2)*(P1/E)*sqrt(1-(P1/E)^2)+...
    +((K1-K2)/2)*(P2/E)*sqrt(1-(P2/E)^2)+...
    +((K2-K3)/2)*(P3/E)*sqrt(1-(P3/E)^2)+...
    +((K3-K4)/2)*(P4/E)*sqrt(1-(P4/E)^2));
elseif E>-P4 && E<=-P3
    N=(4/pi)*(K3*pi/4+((K0-K1)/2)*asin(-P1/E)+((K1-K2)/2)*asin(-
P2/E)+...
    +((K2-K3)/2)*asin(-P3/E)+((K0-K1)/2)*(-P1/E)*sqrt(1-(-
P1/E)^2)+...
    +((K1-K2)/2)*(-P2/E)*sqrt(1-(-P2/E)^2)+...
    +((K2-K3)/2)*(-P3/E)*sqrt(1-(-P3/E)^2));
elseif E<P4 && E>=P3
    N=(4/pi)*(K3*pi/4+((K0-K1)/2)*asin(P1/E)+((K1-
K2)/2)*asin(P2/E)+...
    +((K2-K3)/2)*asin(P3/E)+((K0-K1)/2)*(P1/E)*sqrt(1-
(P1/E)^2)+...
    +((K1-K2)/2)*(P2/E)*sqrt(1-(P2/E)^2)+...
    +((K2-K3)/2)*(P3/E)*sqrt(1-(P3/E)^2));
elseif E>-P3 && E<=-P2
    N=(4/pi)*(K2*pi/4+((K0-K1)/2)*asin(-P1/E)+((K1-K2)/2)*asin(-
P2/E)+ ...
    +((K0-K1)/2)*(-P1/E)*sqrt(1-(-P1/E)^2)+...
    +((K1-K2)/2)*(-P2/E)*sqrt(1-(-P2/E)^2));
elseif E<P3 && E>=P2
    N=(4/pi)*(K2*pi/4+((K0-K1)/2)*asin(P1/E)+((K1-
K2)/2)*asin(P2/E)+ ...
    +((K0-K1)/2)*(P1/E)*sqrt(1-(P1/E)^2)+...
    +((K1-K2)/2)*(P2/E)*sqrt(1-(P2/E)^2));
elseif E>-P2 && E<=-P1
    N=(2*K1/pi)*(pi/2+((K0/K1)-1)*(asin(-P1/E)+...
    +((-P1/E)*sqrt(1-(-P1/E)^2))));
elseif E<P2 && E>=P1
    N=(2*K1/pi)*(pi/2+((K0/K1)-1)*(asin(P1/E)+...
    +((P1/E)*sqrt(1-(P1/E)^2))));
else
    N = K0;
end
E;
A(n,1)=E;
A(n,2)=N;
end
A
t=1:101;
x=A(t,2);
plot(A(:,1),A(:,2));
grid;
title('Plot of Describing function N(e)against input e -
Polynomial approximation n=4');
xlabel('error e (input to the non-linearity)');

```

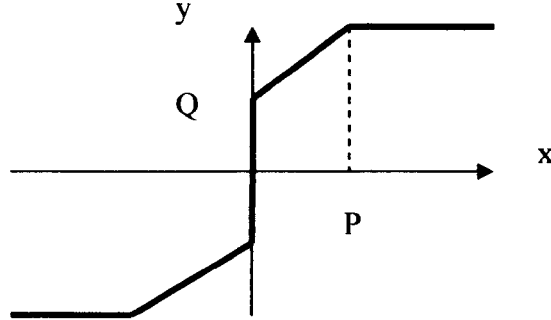


```
ylabel('Describing function output  $N(e)$ ')
```

APPENDIX 1.3

DESCRIBING FUNCTION FOR COULOMB FRICTION PLUS VISCOUS DRAG PLUS SATURATION

Fig. A1.3.1: Characteristic for Coulomb friction + viscous drag + saturation



With reference to the Figure 4.2: $K_1 = 0$, $K_2 = K$, $K_3 = 0$

This is similar to the case of Coulomb friction plus viscous drag but the integration has to be split into two parts.

$$N = \frac{4}{\pi X} \int_0^\alpha y \cdot \sin(\omega t) \cdot d(\omega t) + \frac{4}{\pi X} \int_\alpha^{\frac{\pi}{2}} (Q + KP) \cdot \sin(\omega t) \cdot d(\omega t)$$

$$\therefore N = \frac{4}{\pi X} \int_0^\alpha (Q + KX \sin(\omega t)) \cdot \sin(\omega t) \cdot d(\omega t) + \frac{4}{\pi X} \int_\alpha^{\frac{\pi}{2}} (Q + KP) \cdot \sin(\omega t) \cdot d(\omega t)$$

$$\therefore N = \frac{4}{\pi X} \left[\int_0^\alpha Q \cdot \sin(\omega t) \cdot d(\omega t) + \int_0^\alpha KX \sin^2(\omega t) \cdot d(\omega t) + \int_\alpha^{\frac{\pi}{2}} Q \sin(\omega t) \cdot d(\omega t) + \int_\alpha^{\frac{\pi}{2}} KP \sin(\omega t) \cdot d(\omega t) \right]$$

$$\therefore N = \frac{4}{\pi X} \left[\int_0^{\frac{\pi}{2}} Q \cdot \sin(\omega t) \cdot d(\omega t) + \int_0^\alpha KX \sin^2(\omega t) \cdot d(\omega t) + \int_\alpha^{\frac{\pi}{2}} KP \cdot \sin(\omega t) \cdot d(\omega t) \right]$$

$$\therefore N = \frac{4Q}{\pi X} + \frac{4KP}{\pi X} \cdot \cos(\alpha) + \frac{4K}{\pi} \int_0^\alpha \left(\frac{1}{2} - \frac{1}{2} \cos(2\omega t) \right) d(\omega t)$$

$$\therefore N = \frac{4Q}{\pi X} + \frac{4KP}{\pi X} \cdot \cos(\alpha) + \frac{2K}{\pi} \alpha - \frac{K}{\pi} \cdot \sin(2\alpha)$$

$$\text{Now } P = X \cdot \sin(\alpha) \quad \text{so} \quad \alpha = \sin^{-1}\left(\frac{P}{X}\right) \quad \cos(\alpha) = \sqrt{1 - \left(\frac{P}{X}\right)^2}$$

$$\text{and} \quad \sin(2\alpha) = 2 \cdot \frac{P}{X} \sqrt{1 - \left(\frac{P}{X}\right)^2}$$

$$\therefore N = \frac{4Q}{\pi X} + \frac{4KP}{\pi X} \sqrt{1 - \left(\frac{P}{X}\right)^2} + \frac{2K}{\pi} \sin^{-1}\left(\frac{P}{X}\right) - 2 \frac{K}{\pi} \cdot \frac{P}{X} \cdot \sqrt{1 - \left(\frac{P}{X}\right)^2}$$

$$\therefore N = \frac{4Q}{\pi X} + \frac{2KP}{\pi X} \sqrt{1 - \left(\frac{P}{X}\right)^2} + \frac{2K}{\pi} \sin^{-1}\left(\frac{P}{X}\right) \quad (4.13)$$

APPENDIX A2

INVERSE NYQUIST CALCULATIONS

AND

ROOT LOCI

APPENDIX 2.1

INVERSE NYQUIST CALCULATIONS

Matlab Coding

```

1  clear all
2  % Inverse Nyquist Calculations
3  disp('Inverse Nyquist Calculations')
4  disp(' ')
5  disp('      w(rad/s)      real      imag')
6  n=0;
7  for w=2.36:0.01:2.6
8  n=n+1;
9  g=((j*w)^3+5*(j*w)^2+6*(j*w)+1)/50;
10 x=real(g);
11 y=imag(g);
12 A(n,1)=w;
13 A(n,2)=x;
14 A(n,3)=y;
15 end
16 A

```

Relevant results

w(rad/s)	real	imag					
2.3600	-0.5370	0.0203					
2.3700	-0.5417	0.0182					
2.3800	-0.5464	0.0160					
2.3900	-0.5512	0.0138					
2.4000	-0.5560	0.0115					
2.4100	-0.5608	0.0092					
2.4200	-0.5656	0.0070					
2.4300	-0.5705	0.0046					
2.4400	-0.5754	0.0023					
2.4500	-0.5802	-0.0001	←	{	2.4480	-0.5793	0.0004
2.4600	-0.5852	-0.0025			2.4490	-0.5798	0.0001
2.4700	-0.5901	-0.0050			2.4500	-0.5802	-0.0001
2.4800	-0.5950	-0.0075			2.4510	-0.5807	-0.0004
2.4900	-0.6000	-0.0100					
2.5000	-0.6050	-0.0125					
2.5100	-0.6100	-0.0151					
2.5200	-0.6150	-0.0177					
2.5300	-0.6201	-0.0203					
2.5400	-0.6252	-0.0229					
2.5500	-0.6302	-0.0256					

Nyquist value at crossover when gain is 50 is 0.58 ± 0.001

Inverse Nyquist when gain is increased to 70

Inverse Nyquist Calculations

w(rad/s) real imag

A =

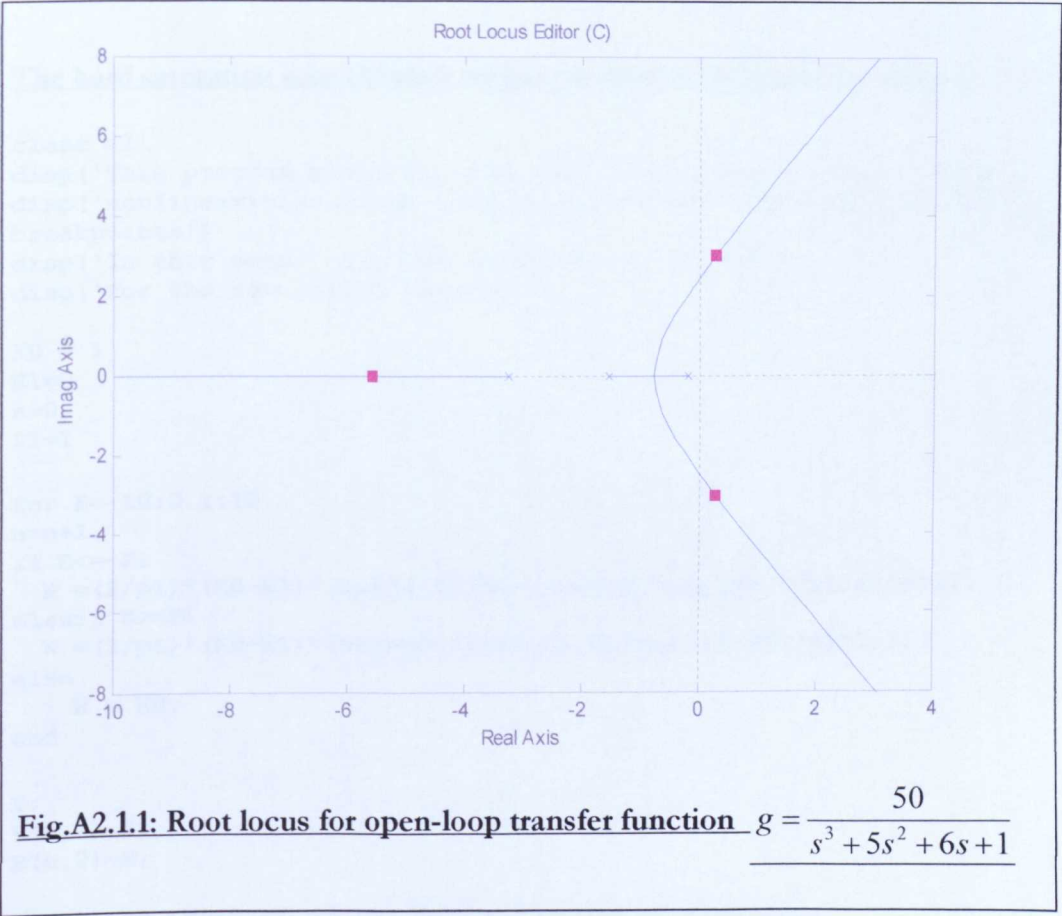
2.4000	-0.3971	0.0082	
2.4100	-0.4006	0.0066	
2.4200	-0.4040	0.0050	
2.4300	-0.4075	0.0033	
2.4400	-0.4110	0.0016	
2.4500	-0.4145	-0.0001	←
2.4600	-0.4180	-0.0018	
2.4700	-0.4215	-0.0036	
2.4800	-0.4250	-0.0053	
2.4900	-0.4286	-0.0071	
2.5000	-0.4321	-0.0089	

Nyquist value at crossover when gain is 70 is 0.41 ± 0.05

Matlab code same as in the first case except line 9 changes to:

```
9      g=( (j*w)^3+5*(j*w)^2+6*(j*w)+1)/70;
```

ROOT LOCUS



APPENDIX 2.2

CODING FOR SELECTED DESCRIBING FUNCTIONS

The hard saturation case (Matlab coding: modified program Gensolh.m)

```

clear all
disp('This program plots out the real describing function for')
disp('nonlinearities which only have straight lines between the
breakpoints')
disp('In this case, only one break point, so n=1')
disp('for the saturation function')

K0 = 1
K1=0
n=0
P1=1

for E=-10:0.1:10
n=n+1
if E<=-P1
    N =(2/pi)*(K0-K1)*(asin(-P1/E)+((-P1/E)*sqrt(1-(-P1/E)^2)));
elseif E>=P1
    N =(2/pi)*(K0-K1)*(asin(P1/E)+((P1/E)*sqrt(1-(P1/E)^2)));
else
    N = K0;
end

E;
B(n,1)=E;
B(n,2)=N;

% C gives the plot of the Nyquist value at crossover

C(n,1)=E;
C(n,2)=0.57;

% D gives the magnitude of the limit-cycle

D(n,1)=2.15;
D(n,2)=1.1*N-0.02;
end
B
C
D
plot(B(:,1),B(:,2),C(:,1),C(:,2),'r',D(:,1),D(:,2),'k','Linewidth'
,1.5);
grid;
title('Fig.5.4:Plot of Describing function N(x)against input x -
Hard Saturation');
xlabel('input to the non-linearity');
ylabel('Describing function output N(x)')

```

```
legend('Describing function','Nyquist value at
crossover','Magnitude of limit-cycle')
```

The soft saturation case (Gensols.m)

```
clear all
disp('This program plots out the real describing function for')
disp('the soft saturation case. This has straight lines between')
disp('the breakpoints. In this case, only one break point, so n=1')
disp('with intial slope 1 and final slope 0.1')
K0 = 1
K1=0.1
n=0
P1=1
for E=-10:0.1:10
n=n+1
if E<=-P1
    N = (2/pi)*(K0-K1)*(asin(-P1/E)+((-P1/E)*sqrt(1-(-P1/E)^2)));
elseif E>=P1
    N = (2/pi)*(K0-K1)*(asin(P1/E)+((P1/E)*sqrt(1-(P1/E)^2)));
else
    N = K0;
end

E;

B(n,1)=E;
B(n,2)=N;
C(n,1)=E;
C(n,2)=0.57;
D(n,1)=1.91;
D(n,2)=1.1*N-0.02;
end
B
C
D
plot(B(:,1),B(:,2),C(:,1),C(:,2),'r',D(:,1),D(:,2),'k','Linewidth'
,1.5);
grid;
title('Fig.5.5:Plot of Describing function N(x)against input x -
Soft Saturation');
xlabel('input to the non-linearity');
ylabel('Describing function output N(x)')
legend('Describing function','Nyquist value at
crossover','Magnitude of limit-cycle')
```

The dead-zone plus hard saturation case (dead_sat_df.m)

```
clear all
disp('This program plots out the describing function for')
disp('dead-zone plus hard saturation')
% Value of slope of the inner linear section
K0 = 0
% Value of the slope in the middle linear section
K1 = 1
```

```
% Value of the slope in the outer linear section')
K2 = 0
% Value at which the first break-point occurs')
P1 = 0.5
% Value at which the second break-point occurs')
P2 = 1.5
n=0;
for E=0:0.1:10
    n=n+1;
    if E>=P2
        N=(4/pi)*(K2*pi/4+((K0-K1)/2)*asin(P1/E)+((K1-
K2)/2)*asin(P2/E)+ ...
        +((K0-K1)/2)*(P1/E)*sqrt(1-(P1/E)^2)+((K1-
K2)/2)*(P2/E)*sqrt(1-(P2/E)^2));
    elseif E<P2 && E>P1
        N=(2*K1/pi)*(pi/2+((K0/K1)-1)*(asin(P1/E)+((P1/E)*sqrt(1-
(P1/E)^2)))));
    else
        N = K0;
    end
    E;
    A(n,1)=E;
    A(n,2)=N;

% C gives the plot of the Nyquist value at crossover

C(n,1)=E;
C(n,2)=0.57;

% D gives the magnitude of the limit cycle

D(n,1)=1.79;
D(n,2)=1.2*N-0.1;
end
A
C
D

plot(A(:,1),A(:,2),C(:,1),C(:,2),'r',D(:,1),D(:,2),'k','Linewidth'
,1.5);
grid;
xlabel('input to the non-linearity');
ylabel('Describing function output N(x)')
legend('Describing function','Nyquist value at
crossover','Magnitude of limit-cycle')
```

The dead-zone plus soft saturation case (dead_softsat.m)

```
clear all
disp('This program plots out the describing function for')
disp('dead-zone plus soft saturation')
% Value of slope of the inner linear section
K0 = 0
% Value of the slope in the middle linear section
```



```

K1 = 1
% Value of the slope in the outer linear section')
K2 = 0.1
% Value at which the first break-point occurs')
P1 = 0.5
% Value at which the second break-point occurs')
P2 = 1.5
n=0;
for E=0:0.1:10
    n=n+1;
    if E>=P2
        N=(4/pi)*(K2*pi/4+((K0-K1)/2)*asin(P1/E)+((K1-
K2)/2)*asin(P2/E)+ ...
        +((K0-K1)/2)*(P1/E)*sqrt(1-(P1/E)^2)+((K1-
K2)/2)*(P2/E)*sqrt(1-(P2/E)^2));
    elseif E<P2 && E>P1
        N=(2*K1/pi)*(pi/2+((K0/K1)-1)*(asin(P1/E)+((P1/E)*sqrt(1-
(P1/E)^2)))));
    else
        N = K0;
    end
    E;
    A(n,1)=E;
    A(n,2)=N;

% C gives the plot of the Nyquist value at crossover

C(n,1)=E;
C(n,2)=0.57;

% D gives the magnitude of the limit cycle

D(n,1)=1.87;
D(n,2)=1.2*N-0.1;
end
A
C
D

plot(A(:,1),A(:,2),C(:,1),C(:,2),'r',D(:,1),D(:,2),'k','Linewidth'
,1.5);
grid;
xlabel('input to the non-linearity');
ylabel('Describing function output N(x)');
legend('Describing function','Nyquist value at
crossover','Magnitude of limit-cycle')

```

The Coulomb friction or ideal relay case (Coulomb.m)

```

clear all
disp('This program plots out the describing function for')
disp('a simple Coulomb friction non-linearity')

%'Q is the value of the cut-off on the y-axis')

```

```
Q = 1;
n=0;
for E=0:0.1:10
    n=n+1;
    if E>=0.1
        N=(4*Q/(pi*E));
    elseif E<=0.1
        N=(4*Q/(pi*E));
    else
        N=0;
    end
    E;
    A(n,1)=E;
    A(n,2)=N;

% C gives the plot of the Nyquist values at crossover

C(n,1)=E;
C(n,2)=0.57;

% D gives the magnitude of the limit-cycle

D(n,1)=2.25;
D(n,2)=1.1*N-0.02;
end
A;
C;
D;
plot(A(:,1),A(:,2),C(:,1),C(:,2),'r',D(:,1),D(:,2),'k','Linewidth'
,1.5);
grid;
xlabel('x (input to the non-linearity)');
ylabel('Describing function output N(x)');
legend('Describing function','Nyquist value at
crossover','Magnitude of limit-cycle')
```

Coulomb friction plus viscous drag (Coulombviscous.m)

```
clear all
disp('This program plots out the describing function for')
disp('a simple Coulomb friction + viscous drag non-linearity')

% Q is the value of the cut-off on y-axis

Q = 1;

% K is the value of the slope the viscous drag

K = 0.3;
n=0;
for E=2:0.1:10
    n=n+1;
```

```

if E>=0.1
    N=(4*Q/(pi*E))+ K;
else
    N=0;
end
E;
A(n,1)=E;
A(n,2)=N;
C(n,1)=E;
C(n,2)=0.57;
D(n,1)=4.72;
D(n,2)=1.18*N-0.2;
end
A;
C;
D;

plot(A(:,1),A(:,2),C(:,1),C(:,2), 'r',D(:,1),D(:,2), 'k', 'Linewidth'
,1.5);
grid;
% title('Plot of Describing function N(x) against input x - Coulomb
friction + Viscous Drag non-linearity');
xlabel('input x');
ylabel('Describing function output N(x)');
legend('Describing function','inverse Nyquist value at
crossover','Magnitude of limit-cycle')

```

Coulomb friction plus viscous drag plus saturation (Coulomb viscous sat.m)

```

clear all
disp('This program plots out the describing function for')
disp('a simple Coulomb friction + viscous drag')
disp('+ saturation non-linearity')
% Q is the value of the cut-off on y-axis
Q = 1;
% K is the value of the slope the viscous drag
K = 0.4;
% P is the value on the x-axis at which the saturation starts')
P=1.5;
n=0;
for E=1:0.1:10
    n=n+1;
    if E>1.5
        N=(4*Q/(pi*E))+((2*K*P)/(pi*E))*sqrt(1-
(P/E)^2)+((2*K)/pi)*asin(P/E);
    elseif E<=1.5 && E>=0.1
        N=(4*Q)/(pi*E)+K;
    else
        N=0;
    end
    E;
    A(n,1)=E;
    A(n,2)=N;
    C(n,1)=E;

```

```

C(n,2)=0.57;
D(n,1)=3.53;
D(n,2)=1.3*N-0.2;
end
A;
C;
D;

plot(A(:,1),A(:,2),C(:,1),C(:,2),'r',D(:,1),D(:,2),'k','Linewidth'
,1.75);
grid;
% title('Plot of Describing function N(x) against input x - Coulomb
friction + Viscous Drag non-linearity');
xlabel('input x');
ylabel('Describing function output N(x)');
legend('Describing function','inverse Nyquist value at
crossover','Magnitude of limit-cycle')

```

Triple-slope characteristic – 3 cases (triple_slope.m)

```

clear all
disp('This program plots out the describing function for')
disp('a polynomial approximation with two break points')
disp('Please state the value of slope of the inner linear section')
K0 = input('K0 = ')
disp('Please state the value of the slope in the middle linear
section')
K1 = input('K1 = ')
disp('Please state the value of the slope in the outer linear
section')
K2 = input('K2 = ')
disp('Please state the value at which the first break-point
occurs')
P1 = input('P1 = ')
disp('Please state the value at which the second break-point
occurs')
P2 = input('P2 = ')
n=0;
for E=0:0.1:10
    n=n+1;
    if E<=-P2
        N=(4/pi)*(K2*pi/4+((K0-K1)/2)*asin(-P1/E)+((K1-K2)/2)*asin(-
P2/E)+ ...
        +((K0-K1)/2)*(-P1/E)*sqrt(1-(-P1/E)^2)+((K1-K2)/2)*(-
P2/E)*sqrt(1-(-P2/E)^2));
    elseif E>=P2
        N=(4/pi)*(K2*pi/4+((K0-K1)/2)*asin(P1/E)+((K1-
K2)/2)*asin(P2/E)+ ...
        +((K0-K1)/2)*(P1/E)*sqrt(1-(P1/E)^2)+((K1-
K2)/2)*(P2/E)*sqrt(1-(P2/E)^2));
    elseif E>-P2 && E<P1
        N=(2*K1/pi)*(pi/2+((K0/K1)-1)*(asin(-P1/E)+((-P1/E)*sqrt(1-(-
P1/E)^2)))));
    elseif E<P2 && E>P1

```

```

        N=(2*K1/pi)*(pi/2+((K0/K1)-1)*(asin(P1/E)+((P1/E)*sqrt(1-
(P1/E)^2)))));
    else
        N = K0;
    end
    E;
    A(n,1)=E;
    A(n,2)=N;
    C(n,1)=E;
    C(n,2)=0.57;
    D(n,1)=8.55;
    D(n,2)=1.3*N-0.2;

end
A
C
D

plot(A(:,1),A(:,2),C(:,1),C(:,2),'r',D(:,1),D(:,2),'k','Linewidth'
,1.5);
grid;
xlabel('input to the non-linearity');
ylabel('Describing function output N(x)');
legend('Describing function','Nyquist value at
crossover','Magnitude of limit-cycle')
title('K0=0.4,K1=1.0,K2=0.2,P1=1,P2=4')

```

Non-linearity with three break-points - first case with more than one limit-cycle (three breakpoints.m)

```

%Describing function of a non-linearity with an inner slope K0,
%a middle slope2 K1 and K2 and an outer slope K3.
%Break points at P1, P2 and P3.

clear all
disp('This program plots out the describing function for')
disp('a polynomial approximation with three break points')
disp('Please state the value of slope of the inner linear section')
K0 = input('K0 = ')
disp('Please state the value of the slope in the first middle
linear section')
K1 = input('K1 = ')
disp('Please state the vaue of the next section')
K2 = input('K2 = ')
disp('Please state the value of the slope in the outer linear
section')
K3 = input('K3 = ')
disp('Please state the value at which the first break-point
occurs')
P1 = input('P1 = ')
disp('Please state the value at which the second break-point
occurs')
P2 = input('P2 = ')

```

```

disp('Please state the value at which the third break-point
occurs')
P3 = input('P3 = ')
n=0;
for E=0:0.1:10
    n=n+1;
    %if E<=-P3
    % N=(4/pi)*(K3*pi/4+((K0-K1)/2)*asin(-P1/E)+((K1-K2)/2)*asin(-
P2/E)+...
    % +((K2-K3)/2)*asin(-P3/E)+((K0-K1)/2)*(-P1/E)*sqrt(1-(-
P1/E)^2)+...
    % +((K1-K2)/2)*(-P2/E)*sqrt(1-(-P2/E)^2)+((K2-K3)/2)*(-
P3/E)*sqrt(1-(-P3/E)^2));
    %else
    if E>=P3
        N=(4/pi)*(K3*pi/4+((K0-K1)/2)*asin(P1/E)+((K1-
K2)/2)*asin(P2/E)+...
        +((K2-K3)/2)*asin(P3/E)+((K0-K1)/2)*(P1/E)*sqrt(1-
(P1/E)^2)+...
        +((K1-K2)/2)*(P2/E)*sqrt(1-(P2/E)^2)+((K2-
K3)/2)*(P3/E)*sqrt(1-(P3/E)^2));
    %elseif E>-P3 && E<-P2
    % N=(4/pi)*(K2*pi/4+((K0-K1)/2)*asin(-P1/E)+((K1-K2)/2)*asin(-
P2/E)+ ...
    % +((K0-K1)/2)*(-P1/E)*sqrt(1-(-P1/E)^2)+((K1-K2)/2)*(-
P2/E)*sqrt(1-(-P2/E)^2));
    elseif E<P3 && E>P2
        N=(4/pi)*(K2*pi/4+((K0-K1)/2)*asin(P1/E)+((K1-
K2)/2)*asin(P2/E)+ ...
        +((K0-K1)/2)*(P1/E)*sqrt(1-(P1/E)^2)+((K1-
K2)/2)*(P2/E)*sqrt(1-(P2/E)^2));
    %elseif E>-P2 && E<-P1
    % N=(2*K1/pi)*(pi/2+((K0/K1)-1)*(asin(-P1/E)+((-P1/E)*sqrt(1-(-
P1/E)^2)))));
    elseif E<=P2 && E>P1
        N=(2*K1/pi)*(pi/2+((K0/K1)-1)*(asin(P1/E)+((P1/E)*sqrt(1-
(P1/E)^2)))));
    else
        N = K0;
    end
    E;
    A(n,1)=E;
    A(n,2)=N;

    % C gives the plot of the Nyquist value ay crossover

    C(n,1)=E;
    C(n,2)=0.57;

    % D gives the magnitude of the FIRST limit-cycle

    D(n,1)=1.85;
    D(n,2)=1.2*N-0.1;

    % F gives the magnitude of the SECOND limit-cycle

```



```

F(n,1)=8.3;
F(n,2)=1.2*N-0.1;
end
A
C
D
F
plot(A(:,1),A(:,2),C(:,1),C(:,2),'r',D(:,1),D(:,2),'k',F(:,1),F(:,2),'m','Linewidth',1.5);
grid;
ylabel('Output');
xlabel('error e (input to the non-linearity)');
title('K0=1,K1=-0.25,K2=1.5,K3=-0.5,P1=1,P2=3,P3=4');
legend('describing function','magnitude of inverse Nyquist at crossover','first limit-cycle','second limit-cycle')

```

Non-linearity with four break-points, showing possibility of two breakpoints and the possible existence of a critical point

```

% Describing function of a non-linearity with an inner slope K0,
% a middle slopes K1, K2 and K3 and an outer slope K4.
% Break points at P1, P2, P3 and P4.
clear all
disp('This program plots out the describing function for')
disp('a polynomial approximation with FOUR break points')
disp('Please state the value of slope of the inner linear section')
K0 = input('K0 = ')
disp('Please state the value of the slope in the first middle linear section')
K1 = input('K1 = ')
disp('Please state the value of the slope of the next section')
K2 = input('K2 = ')
disp('Please state the value of the slope of the next section')
K3 = input('K3 = ')
disp('Please state the value of the slope of the outer linear section')
K4 = input('K4 = ')
disp('Please state the value at which the first break-point occurs')
P1 = input('P1 = ')
disp('Please state the value at which the second break-point occurs')
P2 = input('P2 = ')
disp('Please state the value at which the third break-point occurs')
P3 = input('P3 = ')
disp('Please state the value at which the fourth break-point occurs')
P4 = input('P4 = ')
n=0;

```

```

for E=0:0.1:10
    n=n+1;
% if E<=-P4
%   N=(4/pi)*(K4*pi/4+((K0-K1)/2)*asin(-P1/E)+((K1-K2)/2)*asin(-
P2/E)+...
%   +((K2-K3)/2)*asin(-P3/E)+((K3-K4)/2)*asin(-P4/E)...
%   +((K0-K1)/2)*(-P1/E)*sqrt(1-(-P1/E)^2)+...
%   +((K1-K2)/2)*(-P2/E)*sqrt(1-(-P2/E)^2)+...
%   +((K2-K3)/2)*(-P3/E)*sqrt(1-(-P3/E)^2)+...
%   +((K3-K4)/2)*(-P4/E)*sqrt(1-(-P4/E)^2));
% else
if E >=P4
    N=(4/pi)*(K4*pi/4+((K0-K1)/2)*asin(P1/E)+((K1-
K2)/2)*asin(P2/E)+...
    +((K2-K3)/2)*asin(P3/E)+((K3-K4)/2)*asin(P4/E)+...
    +((K0-K1)/2)*(P1/E)*sqrt(1-(P1/E)^2)+...
    +((K1-K2)/2)*(P2/E)*sqrt(1-(P2/E)^2)+...
    +((K2-K3)/2)*(P3/E)*sqrt(1-(P3/E)^2)+...
    +((K3-K4)/2)*(P4/E)*sqrt(1-(P4/E)^2));
% elseif E>-P4 && E<=-P3
%   N=(4/pi)*(K3*pi/4+((K0-K1)/2)*asin(-P1/E)+((K1-K2)/2)*asin(-
P2/E)+...
%   +((K2-K3)/2)*asin(-P3/E)+((K0-K1)/2)*(-P1/E)*sqrt(1-(-
P1/E)^2)+...
%   +((K1-K2)/2)*(-P2/E)*sqrt(1-(-P2/E)^2)+...
%   +((K2-K3)/2)*(-P3/E)*sqrt(1-(-P3/E)^2));
elseif E<P4 && E>=P3
    N=(4/pi)*(K3*pi/4+((K0-K1)/2)*asin(P1/E)+((K1-
K2)/2)*asin(P2/E)+...
    +((K2-K3)/2)*asin(P3/E)+((K0-K1)/2)*(P1/E)*sqrt(1-(P1/E)^2)+...
    +((K1-K2)/2)*(P2/E)*sqrt(1-(P2/E)^2)+...
    +((K2-K3)/2)*(P3/E)*sqrt(1-(P3/E)^2));
% elseif E>-P3 && E<=-P2
%   N=(4/pi)*(K2*pi/4+((K0-K1)/2)*asin(-P1/E)+((K1-K2)/2)*asin(-
P2/E)+...
%   +((K0-K1)/2)*(-P1/E)*sqrt(1-(-P1/E)^2)+...
%   +((K1-K2)/2)*(-P2/E)*sqrt(1-(-P2/E)^2));
elseif E<P3 && E>=P2
    N=(4/pi)*(K2*pi/4+((K0-K1)/2)*asin(P1/E)+((K1-
K2)/2)*asin(P2/E)+...
    +((K0-K1)/2)*(P1/E)*sqrt(1-(P1/E)^2)+...
    +((K1-K2)/2)*(P2/E)*sqrt(1-(P2/E)^2));
% elseif E>-P2 && E<=-P1
%   N=(2*K1/pi)*(pi/2+((K0/K1)-1)*(asin(-P1/E)+...
%   +((-P1/E)*sqrt(1-(-P1/E)^2))));
elseif E<P2 && E>=P1
    N=(2*K1/pi)*(pi/2+((K0/K1)-1)*(asin(P1/E)+...
    +((P1/E)*sqrt(1-(P1/E)^2))));
else
    N = K0;
end
E;
A(n,1)=E;
A(n,2)=N;

% C gives the plot of the inverse Nyquist value at crossover

```

```
C(n,1)=E;
C(n,2)=0.57;

% D gives the magnitude of the FIRST limit-cycle

D(n,1)=1.65;
D(n,2)=1.2*N-0.1;

% F gives the magnitude of the SECOND limit-cycle

F(n,1)=5.15;
F(n,2)=1.2*N-0.1;

% G is a critical point

G(n,1)=9.7;
G(n,2)=1.2*N-0.1;
end
A
% t=1:101;
% x=A(t,2);
plot(A(:,1),A(:,2),C(:,1),C(:,2),'r',D(:,1),D(:,2),'k',F(:,1),F(:,2),
'm',G(:,1),G(:,2),':', 'Linewidth',1.5);
grid;
title('K0=1, K1=-0.5, K2=1.5, K3=-1.2,K4=1.5, P1=1,
P2=2,P3=4,P4=6');
xlabel('input x');
ylabel('Describing function output N(x)')
legend('Describing function','Magitude of the inverse Nyquist at
crossover','FIRST limit-cycle','SECOND limit-cycle','critical
point')
```

APPENDIX A3

INVERSE FUNCTIONS

AND

SAMPLE SIMULINK MODELS

APPENDIX 3.1

Relation between the slope of a function and its mirror image

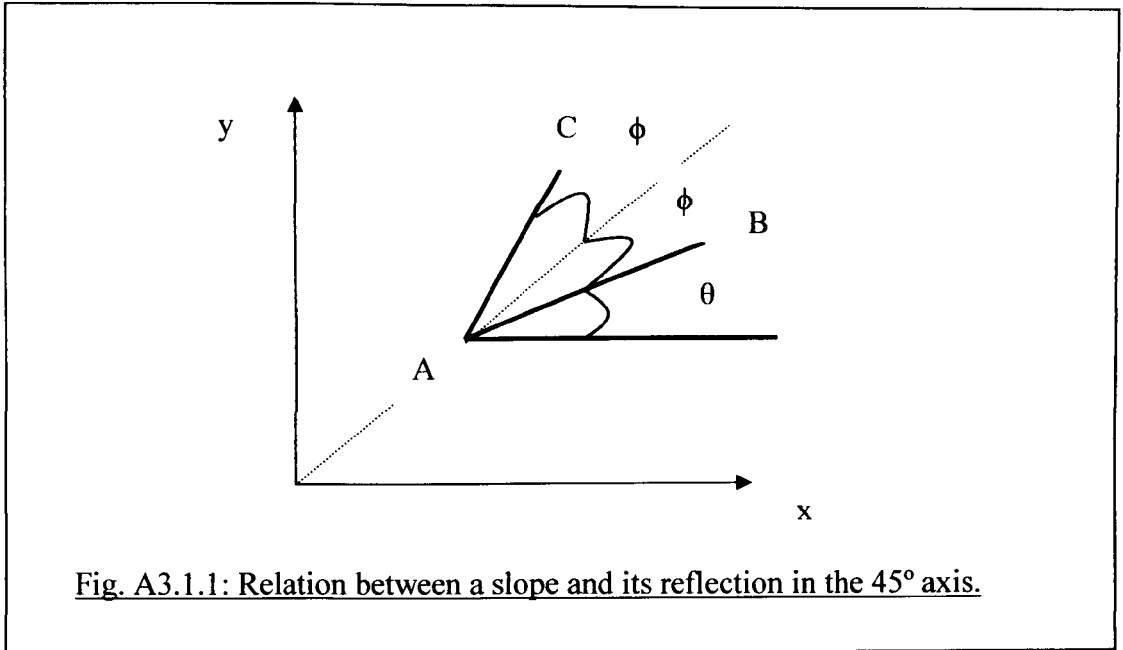


Fig. A3.1.1: Relation between a slope and its reflection in the 45° axis.

Let m be the slope line AB , then $m = \tan(\alpha)$

From the diagram, $\theta + \phi = 45$, so $\phi = 45 - \theta$

If K^* is the slope of line AC , the mirror of AB in the 45° axis,

$$\text{then } K^* = \tan(\theta + 2\phi) = \tan(90 - \theta) = \frac{\tan(90) - \tan(\theta)}{1 + \tan(90) \cdot \tan(\theta)}$$

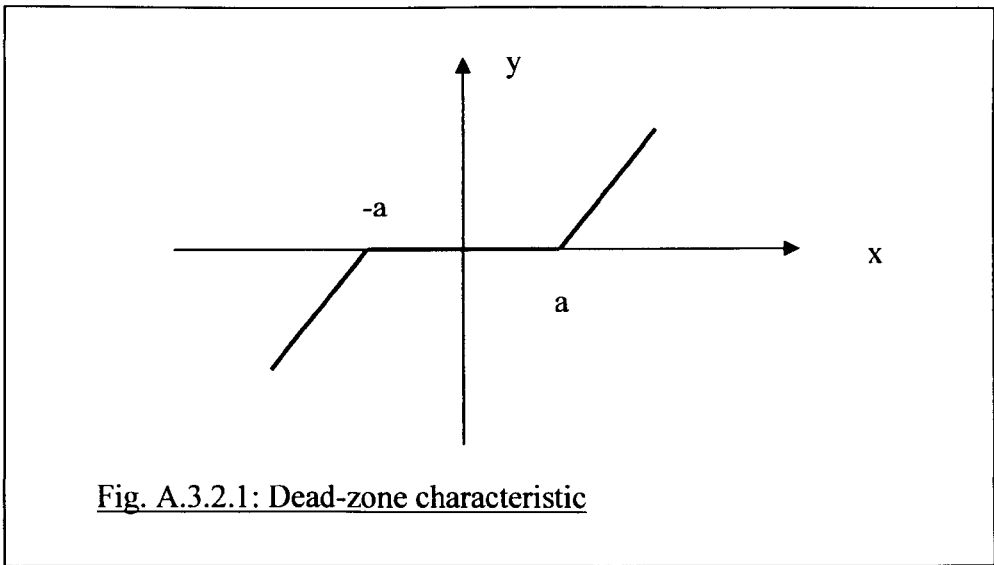
$$\therefore K^* = \frac{1 - \tan(\theta) / \tan(90)}{1 / \tan(90) + \tan(\theta)} = \frac{1}{\tan(\theta)} = \frac{1}{K}$$

$$\therefore K \cdot K^* = 1$$

APPENDIX 3.2

Algebraic relation between dead-zone and Coulomb friction.

This is from unpublished work of Dr Barry Gomm. It shows an algebraic relationship between dead-zone and Coulomb friction. Its significance only became apparent as this research progressed.



$$y = f(x) = \begin{cases} mx + c_1 & x \geq a \\ 0 & -a < x \leq a \\ mx + c_2 & x \leq -a \end{cases} \quad \begin{matrix} m(x-a) \\ \\ m(x+a) \end{matrix}$$

$$\text{when } x = a, \quad y = 0 = mx + c_1 \rightarrow c_1 = -ma$$

$$\text{when } x = -a, \quad y = 0 = mx + c_2 \rightarrow c_2 = +ma$$

$$\text{let } x = g(\omega) = \begin{cases} \omega + a & \omega \geq 0 \\ \omega - a & \omega < 0 \end{cases}$$

Table A3.2.1

Removing dead-zone

w	x	y
$2a$	$3a$	$2am$
a	$2a$	ma
0	a	0
$-a$	$-2a$	$-ma$
$-2a$	$-3a$	$-2am$

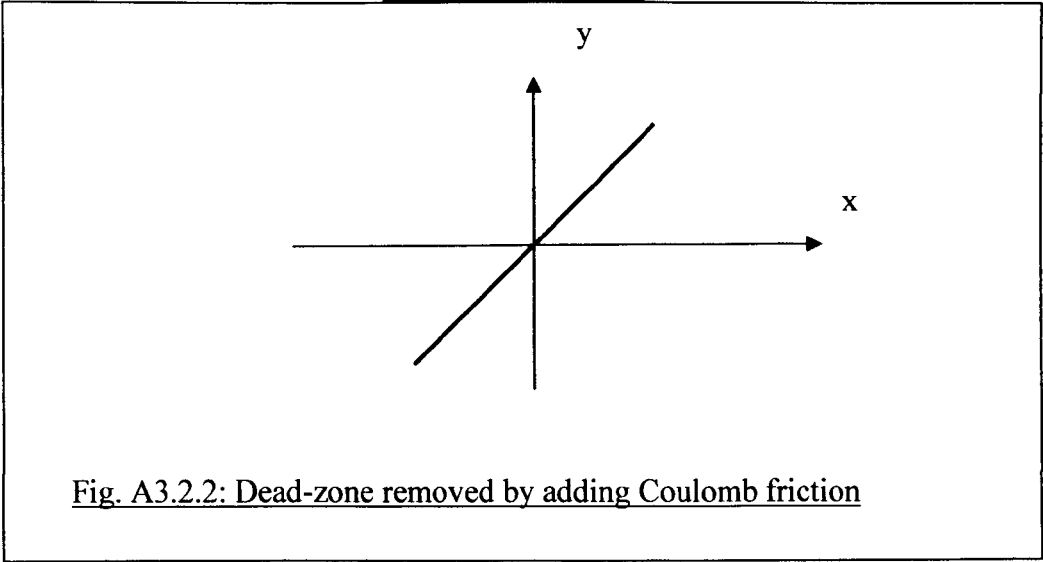


Fig. A3.2.2: Dead-zone removed by adding Coulomb friction

Hence by adding a constant in the y direction, the dead-zone in the x -direction has been removed.

APPENDIX 3.3

The Design of a Fuzzy Non-linearity and its Inverse

Relation between P and P*

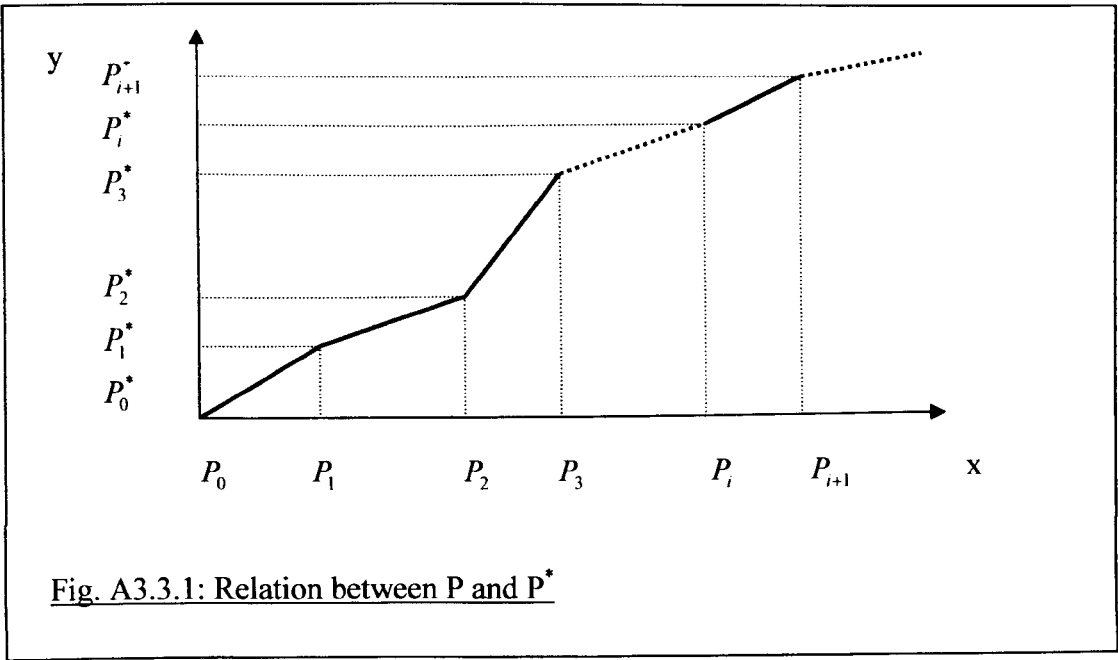


Fig. A3.3.1 shows the relationship between P and P^* for all P_i . If the slope of the line between any P_i and P_{i+1} is K_i it follows that:

$$\begin{aligned} P_1^* &= K_0 P_1 \\ P_2^* &= K_0 P_1 + K_1 (P_2 - P_1) = (K_0 - K_1) P_1 + K_1 P_2 \\ P_3^* &= K_0 P_1 + K_1 (P_2 - P_1) + K_2 (P_3 - P_2) = (K_0 - K_1) P_1 + (K_1 - K_2) P_2 + K_2 P_3 \\ &\vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \\ P_n^* &= (K_0 - K_1) P_1 + (K_1 - K_2) P_2 + \dots + (K_{i-1} - K_i) P_i + \dots + (K_{n-2} - K_{n-1}) P_{n-1} + K_{n-1} P_n \\ \therefore P_n^* &= \sum_{i=1}^{n-1} (K_{i-1} - K_i) P_i + K_{n-1} P_n \dots\dots\dots (A3.3.1) \end{aligned}$$

Relation between c, K and P

Fig. A.3.3.2 shows the output from a type 1 Sugeno fuzzy controller. Each straight section can be represented by the equation $y = Kx + c$.

The equation for the first section is $y = K_0 x + c_0$, where K_0 is the gradient of that section. Since this line passes through the origin, $c_0 = 0$.

At the upper end of the first straight section, start of the second section:

$$y_1 = K_0 P_1 \quad \text{and} \quad x_1 = P_1$$

The equation for the second section is:

$$y_1 = K_1 x_1 + c_1 \Rightarrow K_0 P_1 = K_1 P_1 + c_1$$

$$\therefore c_1 = (K_0 - K_1)P_1$$

At the upper end of the second section, start of the third section:

$$y_2 = K_0 P_1 + K_1 (P_2 - P_1) \quad \text{and} \quad x_2 = P_2$$

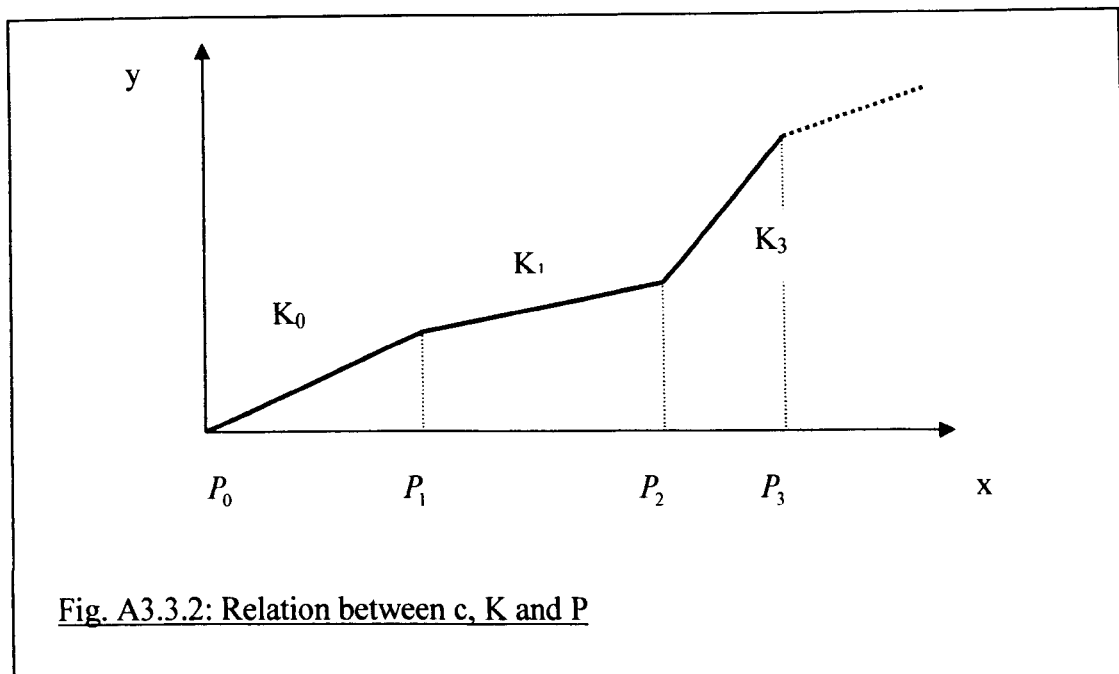
The equation of the second section is:

$$y_2 = K_2 x_2 + c_2 \Rightarrow K_0 P_1 + K_1 (P_2 - P_1) = K_2 P_2 + c_2$$

$$\therefore c_2 = (K_0 - K_1)P_1 + (K_1 - K_2)P_2$$

Repeating this procedure for every section leads to the general result:

$$c_n = \sum_{i=1}^n (K_{i-1} - K_i)P_i \quad \dots\dots\dots (A3.3.2)$$



APPENDIX 3.4

SAMPLE SIMULINK MODELS

SIMULINK open-loop arrangement showing the fuzzy non-linearity in series with its inverse.

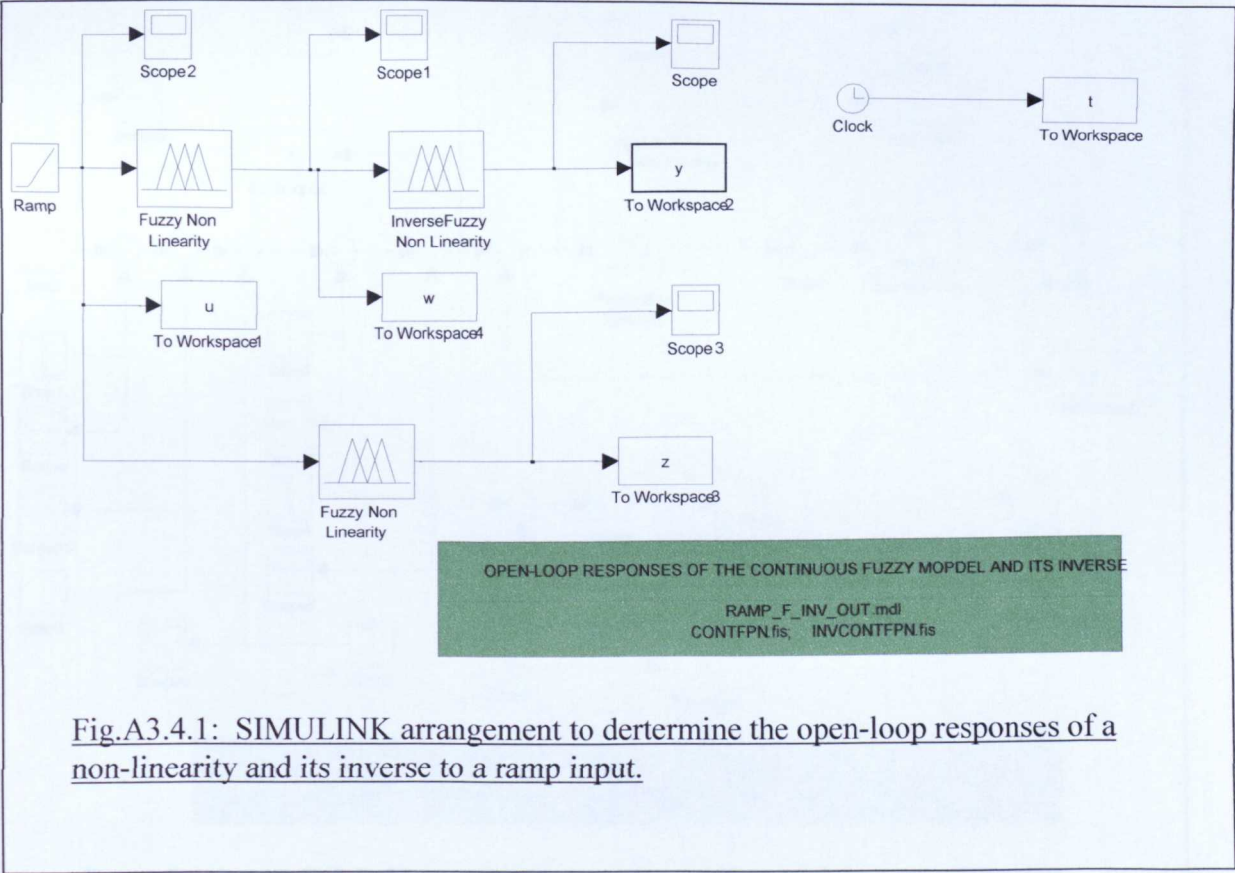
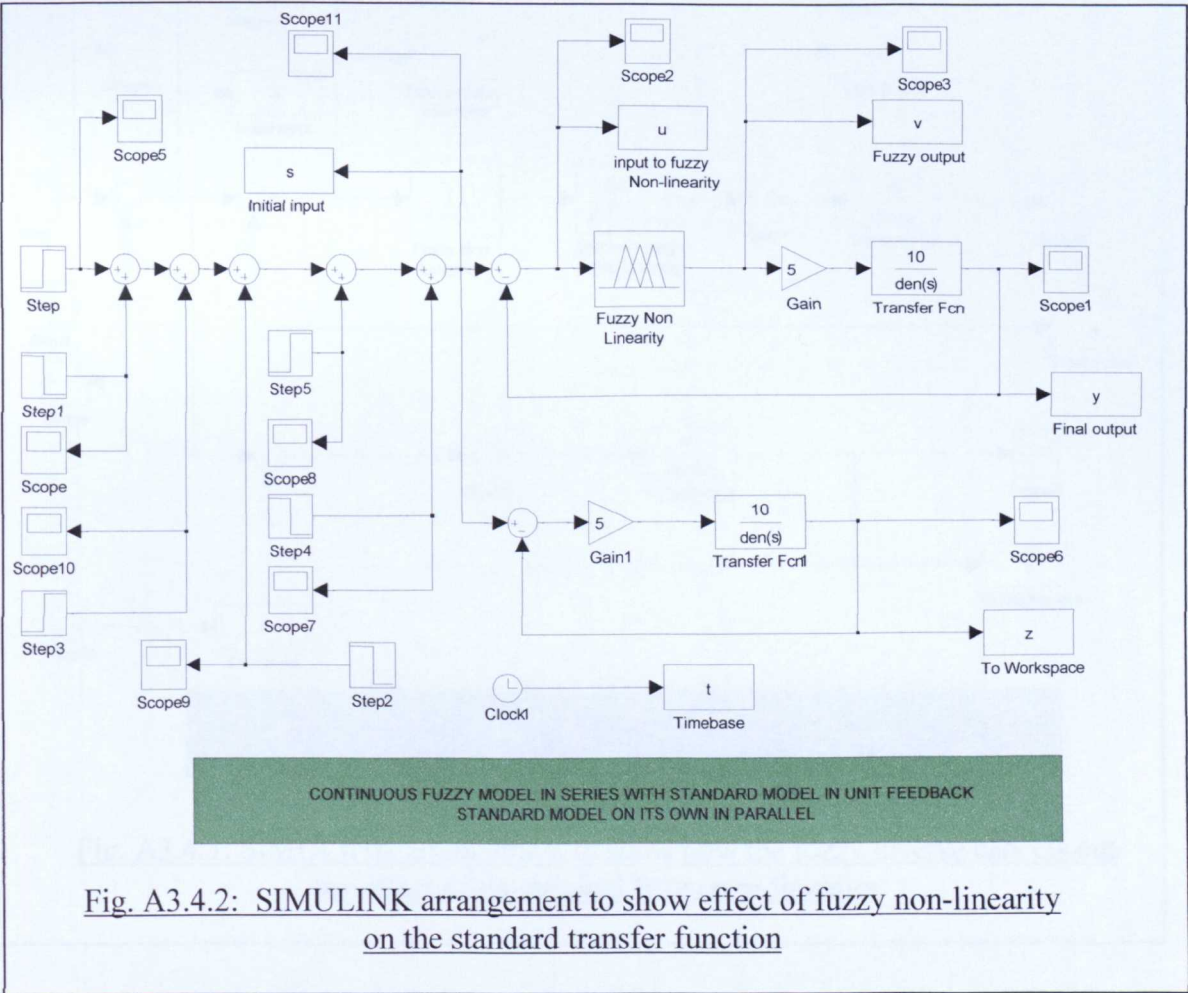
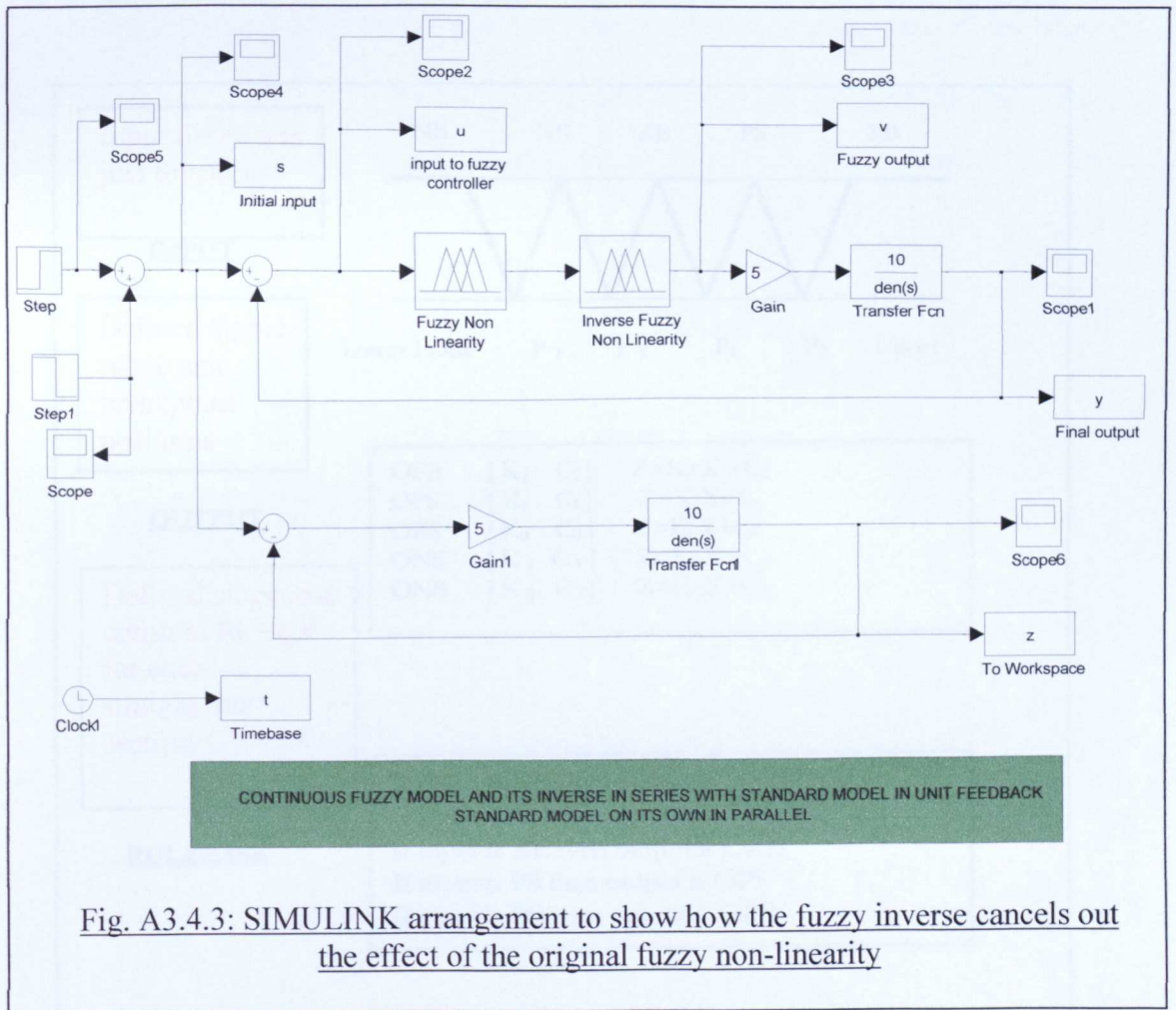


Fig.A3.4.1: SIMULINK arrangement to dertermine the open-loop responses of a non-linearity and its inverse to a ramp input.

SIMULINK arrangement showing the effect of the block, of the fuzzy non-linearity in series in series with the standard tranfer function, with unit feedback. Also present in parallel is the unitary feedback results for the transfer function on its own. Impulse sources at the front end to supply square pulse of varying height to show when various features are triggered, i.e: first and second limit-cycles and the exceeding of the final critical value.

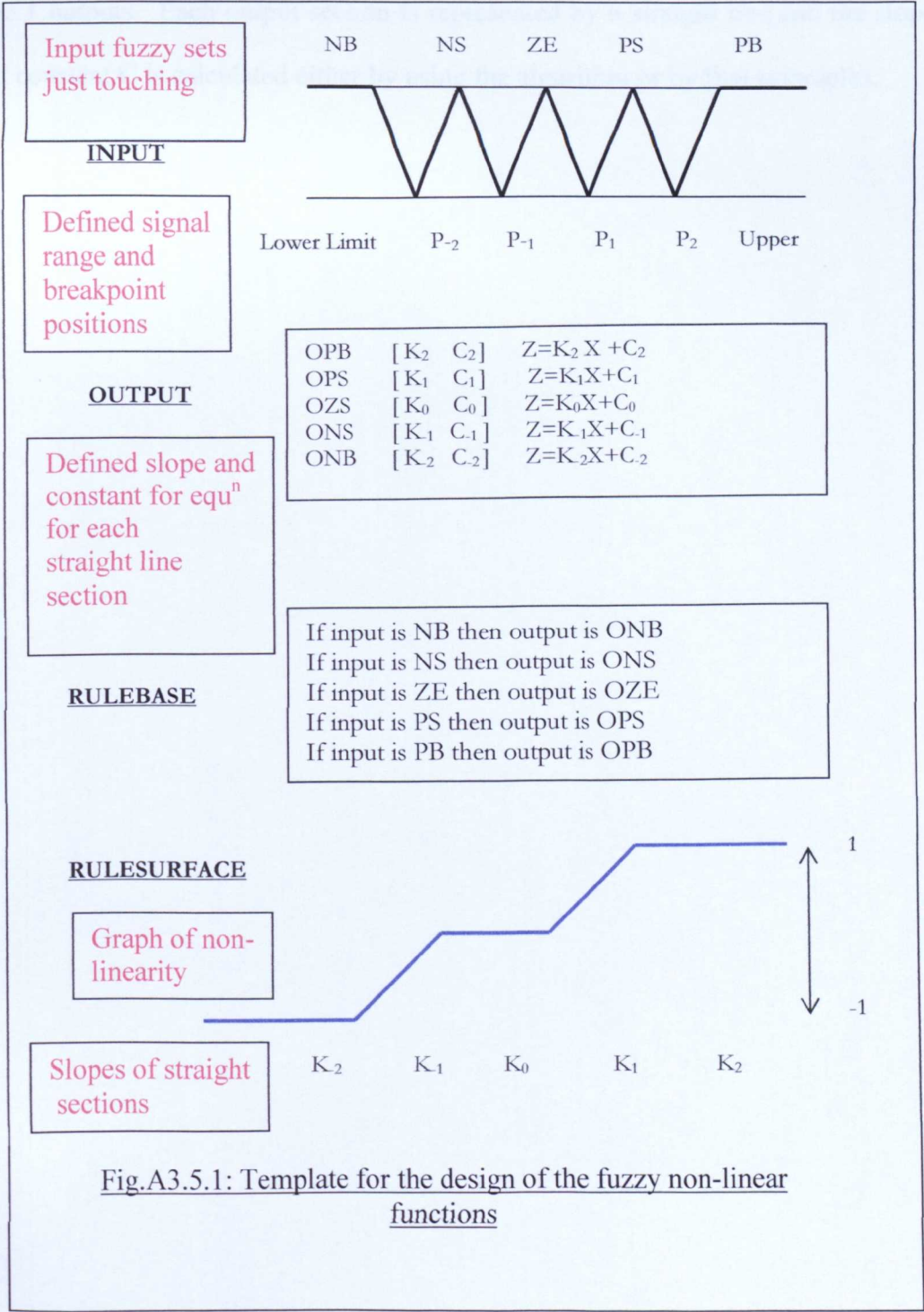


SIMULINK arrangement showing the effect of the block, of the fuzzy non-linearity in series with its inverse in series with the standard transfer function, with unit feedback. Also present in parallel is the unitary feedback results for the transfer function on its own.



APPENDIX A3.5

TEMPLATE FOR THE FUZZY NON-LINEAR FUNCTION DESIGNS



The input to the fuzzy function is via triangular fuzzy sets which are just touching. The boundaries of these fuzzy sets are the break points $P_1 \dots P_n$. These inputs are connected to the linear output functions by a 1-1 rule-base which introduces no distortion since the same break-point values are used as the boundaries of the Sugeno type 1 outputs. Each output section is represented by a straight line and the slope K and constant C is calculated either by using the algorithm or by first principles.